

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE
UNA HERRAMIENTA PARA LA
RESOLUCIÓN DE PROBLEMAS EN EL
ÁMBITO DEL GOBIERNO DE LAS TI
BASADO EN EL ESTÁNDAR COBIT 4.1
ADMINISTRACIÓN**



**UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**

**Proyecto Fin de Carrera
Ingeniería Técnica en Informática de Gestión**

Autor:

Juan Manuel Acedo Ruiz

Profesor:

Anabel Fraga Vázquez

Agradecimientos

A todas las personas que siempre han creído en mí, familia y amigos, en especial a mis padres.

A mi amigo y compañero de proyecto Jorge.

A nuestra tutora Anabel.

Gracias por todo.

Índice

Índice de ilustraciones.....	6
Índice de tablas	10
1. Introducción	13
1.1 Descripción breve del proyecto	13
1.2 Objetivos del proyecto	16
2. Estado del Arte	17
2.1 ¿Qué se conoce como Gobierno TI?	17
2.2 ¿Qué es COBIT?	24
2.3 ¿Qué es ITIL?	32
2.4 ¿Qué es CMMI?	39
2.5 ¿Qué es MOF?	44
2.6 ¿Qué es COSO?	46
2.7 Diferencias de los distintos modelos.....	48
2.8 Certificaciones ISO.....	50
2.9 Otros Conceptos	54
3. Arquitectura Tecnológica.....	61
3.1 Modelo de arquitectura.....	61
3.2 Software utilizado	63
3.3 Datos usados	67
4. Análisis	68
4.1 Análisis de Requisitos	68
4.1.1 Usuarios y roles	68
4.2 Especificación de Requisitos	72
4.2.1 Requisitos funcionales.....	74
4.2.2 Requisitos de rendimiento	78
4.2.3 Requisitos de interfaz.....	78
4.2.4 Requisitos de recursos.....	80
4.2.5 Requisitos de comprobación	80
4.2.6 Requisitos de documentación.....	81
4.2.7 Requisitos de seguridad.....	82
4.2.8 Requisitos de daño	82
4.3 Diagrama de Casos de Uso.....	83
4.3.1 Descripción textual de Casos de Uso del Usuario	85
4.3.2 Descripción textual de Casos de Uso del Administrador	88
4.4 Diagramas de Secuencia	104
5. Diseño Detallado	108
5.1 Diagrama de Clases UML	108
5.1.1 Paquete Sistema Principal	109

5.1.2 Paquete Ficheros	110
5.1.3 Paquete Conexión	111
5.1.4 Paquete Parámetros_SW	112
5.1.5 Paquete Usuario	113
5.1.6 Paquete Usuario_Problema	114
5.1.7 Paquete Usuario_Evaluación	115
5.1.8 Paquete Usuario_COBIT	116
5.1.9 Paquete Solución.....	117
5.1.10 Paquete Administrador	118
5.1.11 Paquete Administrador_BBDD.....	119
5.1.12 Paquete Administrador_Contraseña.....	120
5.1.13 Paquete Administrador_Problema	121
5.1.14 Paquete Administrador_Evaluación.....	122
5.1.15 Paquete Administrador_COBIT	123
5.2 Esquema Entidad-Relación	124
5.3 Descripción detallada	125
5.3.1 Ciclo de Vida.....	126
5.3.2 Interfaz gráfica de usuario	128
5.3.3 Estructura y funcionamiento del código de la aplicación	151
5.3.4 Ofuscación del código.....	221
5.3.5 Inyección SQL	222
5.3.6 Scripts de la base de datos	224
5.3.7 Instalación y ejecución de la aplicación	225
6. Pruebas del Sistema	227
6.1 Objetivo	227
6.2 Especificación del entorno de pruebas	227
6.3 Niveles de pruebas a emplear.....	228
6.4 Plantilla de definición de pruebas.....	228
6.5 Pruebas de caja negra.....	230
6.6 Pruebas de caja blanca	237
7. Análisis Económico	244
7.1 Estimación inicial mediante COCOMO II 2000.....	244
7.1.1 Introducción	244
7.1.2 Grupos de Datos Internos (ILF)	245
7.1.3 Grupos de Datos Externos (ELF)	252
7.1.4 Grupos de Datos Externos (ELF)	253
7.1.5 Salidas Externas (EO)	253
7.1.6 Consultas Externas (EQ).....	254
7.1.7 Cálculo de Puntos de Función y SLOC.....	254
7.1.8 Factores de Escala	255
7.1.9 Drivers de Coste	256
7.1.10 Resultados de la estimación	258
7.2 Costes reales	259
7.2.1 Plan de trabajo: WBS.....	259

7.2.2 Cronograma de planificación	260
7.2.3 Tareas y recursos humanos	264
7.2.4 Informe presupuestario.....	268
8. Líneas futuras	270
9. Conclusiones	273
Anexo A: Glosario.....	274
Anexo B: Bibliografía	278
Anexo C: Referencias.....	279
Anexo D: Manual del usuario	283

Índice de ilustraciones

Ilustración 1. Objetivos de dominios de gobierno de TI	22
Ilustración 2. Estructura de COBIT	26
Ilustración 3. Evaluación del Riesgo	29
Ilustración 4. Ciclo de Vida de ITIL	32
Ilustración 5. Nivel 1 o Nivel Inicial de Madurez	40
Ilustración 6. Modelo MOF	44
Ilustración 7. Cubo COSO con sus áreas conectadas.....	46
Ilustración 8. Mapa de los distintos estándares, metodologías y marcos de trabajo existentes	49
Ilustración 9. Planificación estratégica de la empresa	54
Ilustración 10. Objetivos, métricas, metas y acciones	55
Ilustración 11. Modelo vista-controlador.....	61
Ilustración 12. Casos de Uso de Usuario	83
Ilustración 13. Casos de Uso de Administrador	84
Ilustración 14. Diagrama de Secuencia Buscar por dominio y proceso de COBIT	104
Ilustración 15. Diagrama de Secuencia Cambiar preferencias interfaz	104
Ilustración 16. Diagrama de Secuencia Ayuda.....	105
Ilustración 17. Diagrama de Secuencia Administración problemas	105
Ilustración 18. Diagrama de Secuencia Administración tests de evaluación.....	106
Ilustración 19. Diagrama de Secuencia Administración COBIT	106
Ilustración 20. Diagrama de Secuencia Administración Base de Datos.....	107
Ilustración 21. Diagrama de Secuencia Administración contraseña	107
Ilustración 22. Diagrama de clases a nivel de paquetes.....	108
Ilustración 23. Paquete Sistema Principal	109
Ilustración 24. Paquete Ficheros.....	110
Ilustración 25. Paquete Conexión	111
Ilustración 26. Paquete Parámetros_SW	112
Ilustración 27. Paquete Usuario.....	113
Ilustración 28. Paquete Usuario_Problema.....	114
Ilustración 29. Paquete Usuario_Evaluación	115
Ilustración 30. Paquete Usuario_COBIT	116
Ilustración 31. Paquete Solución	117
Ilustración 32. Paquete Administrador	118
Ilustración 33. Paquete Administrador_BBDD	119
Ilustración 34. Paquete Administrador_Contraseña	120
Ilustración 35. Paquete Administrador_Problema	121
Ilustración 36. Paquete Administrador_Evaluación	122
Ilustración 37. Paquete Administrador_COBIT.....	123
Ilustración 38. Esquema Entidad-Relación	124

Ilustración 39. Logotipo usado en el manual de usuario de ITGS. Registrado como Propiedad Intelectual. Copyright © 2012 Universidad Carlos III de Madrid. Todos los derechos reservados.....	125
Ilustración 40. Diagrama del modelo iterativo incremental	126
Ilustración 41. Estructura de la ventana de la aplicación	130
Ilustración 42. Encabezado de la ventana aplicación con su menú y título.....	130
Ilustración 43. Encabezado de la ventana aplicación con su menú y título en la solución	131
Ilustración 44. Solución de ITGS con tipo de fuente Dialog, fondo blanco y barras de desplazamiento vertical y horizontal (scroll)	132
Ilustración 45. Paleta de colores para la personalización de la barra de menú y botón para cambio de valores por defecto.....	134
Ilustración 46. Menú de Opciones y ventana de Preferencias para personalización del usuario	136
Ilustración 47. Logotipo de la aplicación ITGS Ilustración 48. Logotipo de Java	137
Ilustración 49. Logotipos de ayuda de la aplicación ITGS	137
Ilustración 50. Ejemplo de tooltip. What is process code?	137
Ilustración 51. Accesos mnemónicos. (Alt + F, Alt + S, Alt + P, Alt + E, Ctrl + S, Ctrl + P, Escape)	138
Ilustración 52. Foco en el botón User.....	138
Ilustración 53. El indicador parpadeante Caret Position de Java	139
Ilustración 54. Uso de puntos suspensivos en la barra menú	139
Ilustración 55. Botones de navegación Back (Atrás) y Main menu (Menú principal) en la parte inferior de la ventana.....	140
Ilustración 56. Botones de navegación Back (Atrás) y Main menu (Menú principal) en la parte superior de la ventana	141
Ilustración 57. Botones de navegación Back (Atrás) y Next (Siguiete) en la parte superior de la ventana	142
Ilustración 58. Navegación mediante la barra de menú	142
Ilustración 59. Botones de acción descriptivos y con el mismo tamaño	143
Ilustración 60. Cajas de texto modificables con información previa de fichero	145
Ilustración 61. Caja de texto protegida para contraseñas	145
Ilustración 62. Área de texto con scroll y efecto de palabras enteras.....	146
Ilustración 63. Botones JRadioButton	147
Ilustración 64. Botones JCheckBox	147
Ilustración 65. Lista de procesos COBIT para una determinada solución.....	148
Ilustración 66. Menús desplegables.....	149
Ilustración 67. Etiquetas en la ventana de cambio de contraseña.....	149
Ilustración 68. Ventana de aviso que muestra información del ‘Acerca de...’ de la aplicación	150
Ilustración 69. Árbol con las 44 clases principales del código fuente de la aplicación	152
Ilustración 70. Mensaje del copyright de la aplicación al inicio	155
Ilustración 71. Ventana “Principal” de la aplicación con los perfiles “Usuario” y “Administrador”	157

Ilustración 72. Las tres ramas posibles de consulta del usuario en la ventana	
“Usuario”	158
Ilustración 73. Ventana Cobit (Dominio) con los cuatro dominios de COBIT 4.1	159
Ilustración 74. Ventana Cobit Proceso con los diez procesos relacionados con el	
dominio PO de COBIT 4.1	160
Ilustración 75. Clase y ventana VentanaSolucionCobit.java	161
Ilustración 76. Listado de problemas relacionados en la clase	
ventanaListadoProblemasRelacionados.java desde la rama de COBIT	163
Ilustración 77. Estructura de la ventana ventanaMostrarSolucion	166
Ilustración 78. Clase VentanaMostrarSolucion con el mensaje de proceso y su código	
.....	166
Ilustración 79. Ventana ventanaMostrarSolucion con la solución de COBIT	168
Ilustración 80. Menú File en la ventana ventanaMostrarSolucion	169
Ilustración 81. Salvar en fichero	170
Ilustración 82. Solución en fichero TXT	171
Ilustración 83. Solución en fichero PDF	172
Ilustración 84. Solución en fichero HTML	172
Ilustración 85. Acceso por contraseña del perfil administrador en la ventana	
“Administrador”	173
Ilustración 86. Tareas del administrador	174
Ilustración 87. Parámetros configurables de la base de datos	175
Ilustración 88. Mensaje de error de conexión con la base de datos usando el	
programa con el perfil usuario	176
Ilustración 89. Mensaje de aviso de conexión incorrecta con la base de datos usando	
el perfil administrador	177
Ilustración 90. Mensaje de aviso de conexión correcta establecida con la base de	
datos	178
Ilustración 91. Cambio de la contraseña del administrador	179
Ilustración 92. Altas, bajas y modificación (administración) de información de los	
problemas	182
Ilustración 93. Alta en administración de problemas	185
Ilustración 94. Baja en administración de problemas	187
Ilustración 95. Modificación en administración de problemas	188
Ilustración 96. Modificación de nombres en administración de problemas	191
Ilustración 97. Relación de tablas en la administración de problemas	193
Ilustración 98. Administración de las evaluaciones	194
Ilustración 99. Alta en administración de evaluación	195
Ilustración 100. Baja en administración de evaluaciones	198
Ilustración 101. Modificación en administración de Evaluación	199
Ilustración 102. Modificación de nombre en administración de evaluaciones	201
Ilustración 103. Relación de tablas en la administración de evaluaciones	203
Ilustración 104. Administración de COBIT	204
Ilustración 105. Alta en administración de COBIT	205
Ilustración 106. Baja en administración de COBIT	209
Ilustración 107. Modificación en administración de COBIT	210

Ilustración 108. Cambio de nombres/descripciones en COBIT	212
Ilustración 109. Modelo de madurez en COBIT	215
Ilustración 110. Creado y borrado de relación entre tablas de COBIT	217
Ilustración 111. Vista de la parte superior de las Preferencias del usuario.....	218
Ilustración 112. Vista de la parte inferior de las Preferencias del usuario	219
Ilustración 113. Contenido del fichero comprimido	225
Ilustración 114. Inserción de los scripts en la base de datos	226
Ilustración 115. Cuadro de diálogo para el cálculo de SLOC	255
Ilustración 116: Cuadro de diálogo de los Factores de Escala.....	256
Ilustración 117. Ajuste de los Drivers de Coste	257
Ilustración 118. Resultados de la estimación realizada mediante COCOMO II 2000	258
Ilustración 119. Plan de trabajo: WBS.....	259
Ilustración 120. Diagrama de Gantt – Parte (1/4)	260
Ilustración 121. Diagrama de Gantt – Parte (2/4)	261
Ilustración 122. Diagrama de Gantt – Parte (3/4)	262
Ilustración 123. Diagrama de Gantt – Parte (4/4)	263
Ilustración 124. Tareas y recursos humanos – Parte (1/4).....	264
Ilustración 125. Tareas y recursos humanos – Parte (2/4).....	265
Ilustración 126. Tareas y recursos humanos – Parte (3/4).....	266
Ilustración 127. Tareas y recursos humanos – Parte (4/4).....	267
Ilustración 128. Informe presupuestario – Parte (1/2)	268
Ilustración 129. Informe presupuestario – Parte (2/2)	269

Índice de tablas

Tabla 1. Plantilla genérica de requisitos.....	72
Tabla 2. Requisito software RS-F001.....	74
Tabla 3. Requisito software RS-F002.....	74
Tabla 4. Requisito software RS-F003.....	75
Tabla 5. Requisito software RS-F004.....	75
Tabla 6. Requisito software RS-F005.....	75
Tabla 7. Requisito software RS-F006.....	76
Tabla 8. Requisito software RS-F007.....	76
Tabla 9. Requisito software RS-F008.....	76
Tabla 10. Requisito software RS-F009.....	77
Tabla 11. Requisito software RS-F010.....	77
Tabla 12. Requisito software RS-F011.....	77
Tabla 13. Requisito de rendimiento RS-R001	78
Tabla 14. Requisito de interfaz RS-I001.....	78
Tabla 15. Requisito de interfaz RS-I002.....	79
Tabla 16. Requisito de recurso RS-RE001	80
Tabla 17. Requisito de comprobación RS-C001	80
Tabla 18. Requisito de comprobación RS-C002	81
Tabla 19. Requisito de comprobación RS-C003	81
Tabla 20. Requisito de documentación RS-D001	81
Tabla 21. Requisito de seguridad RS-S001	82
Tabla 22. Requisito de daño RS-DA001	82
Tabla 23. Caso de Uso Gestionar Usuario	85
Tabla 24. Caso de Uso Consultar por COBIT	85
Tabla 25. Caso de Uso Elegir dominio	86
Tabla 26. Caso de Uso Elegir proceso.....	86
Tabla 27. Caso de Uso Elegir problema relacionado.....	86
Tabla 28. Caso de Uso Elegir solución	87
Tabla 29. Caso de Uso Consultar ayuda	88
Tabla 30. Caso de Uso Configurar preferencias.....	88
Tabla 31. Caso de Uso Gestionar Administración.....	88
Tabla 32. Caso de Uso Insertar contraseña de administrador	89
Tabla 33. Caso de Uso Configurar parámetros de la BD	89
Tabla 34. Caso de Uso Introducir datos de conexión de la base de datos.....	89
Tabla 35. Cambiar contraseña de administrador	90
Tabla 36. Caso de Uso Introducir nueva contraseña	90
Tabla 37. Configurar problemas.....	90
Tabla 38. Caso de Uso Dar alta problemas.....	91
Tabla 39. Caso de Uso Introducir datos problemas	91
Tabla 40. Caso de Uso Dar baja problemas	91
Tabla 41. Caso de Uso Seleccionar para borrar problemas.....	92
Tabla 42. Caso de Uso Dar modificación problemas.....	92

Tabla 43. Caso de Uso Cambiar nombres problemas	92
Tabla 44. Introducir nombres problemas.....	93
Tabla 45. Caso de Uso Relacionar tablas problemas	93
Tabla 46. Caso de Uso Crear relación problemas	94
Tabla 47. Caso de Uso Borrar relación problemas.....	94
Tabla 48. Caso de Uso Configurar evaluaciones	94
Tabla 49. Caso de Uso Dar alta evaluaciones	95
Tabla 50. Caso de Uso Introducir datos evaluaciones	95
Tabla 51. Caso de Uso Dar baja evaluaciones	95
Tabla 52. Caso de Uso Seleccionar para borrar evaluaciones	96
Tabla 53. Caso de Uso Dar modificación evaluaciones	96
Tabla 54. Caso de Uso Cambiar nombres evaluaciones.....	97
Tabla 55. Caso de Uso Introducir nombres evaluaciones	97
Tabla 56. Caso de Uso Relacionar tablas evaluaciones.....	98
Tabla 57. Caso de Uso Crear relación evaluaciones.....	98
Tabla 58. Caso de Uso Borrar relación evaluaciones	99
Tabla 59. Caso de Uso Configurar COBIT	99
Tabla 60. Caso de Uso Dar alta COBIT	99
Tabla 61. Caso de Uso Introducir datos COBIT	100
Tabla 62. Caso de Uso Dar baja COBIT	100
Tabla 63. Caso de Uso Seleccionar para borrar COBIT	100
Tabla 64. Caso de Uso Dar modificación COBIT	101
Tabla 65. Caso de Uso Cambiar nombres COBIT.....	101
Tabla 66. Caso de Uso Introducir nombres COBIT	102
Tabla 67. Caso de Uso Relacionar tablas COBIT	102
Tabla 68. Caso de Uso Crear relación COBIT.....	103
Tabla 69. Caso de Uso Borrar relación COBIT	103
Tabla 70. Connotaciones de los colores (Götz (1998) resumido por OpenLearn LabSpace)	133
Tabla 71. Luminosidad de los colores. (Götz (1998) en OpenLearn LabSpace).....	135
Tabla 72. Plantilla para la definición de las Pruebas	228
Tabla 73. PR-NE-001	230
Tabla 74. PR-NE-002	230
Tabla 75. PR-NE-003	231
Tabla 76. PR-NE-004	231
Tabla 77. PR-NE-005	232
Tabla 78. PR-NE-006	232
Tabla 79. PR-NE-007	233
Tabla 80. PR-NE-008	233
Tabla 81. PR-NE-009	234
Tabla 82. PR-NE-010	234
Tabla 83. PR-NE-011	235
Tabla 84. PR-NE-012	235
Tabla 85. PR-NE-013	236
Tabla 86. PR-BL-001	237

Tabla 87. PR-BL-002	237
Tabla 88. BL-NE-003.....	238
Tabla 89. PR-BL-004	238
Tabla 90. PR-BL-005	239
Tabla 91. PR-BL-006	239
Tabla 92. PR-BL-007	240
Tabla 93. PR-BL-008	240
Tabla 94. PR-BL-009	241
Tabla 95. PR-BL-010	241
Tabla 96. PR-BL-011	242
Tabla 97. PR-BL-012	242
Tabla 98. PR-BL-013	243
Tabla 99. Almacén PALABRA	245
Tabla 100. Almacén PALABRA.....	245
Tabla 101. Almacén TIENE	245
Tabla 102. Almacén PROBLEMA	246
Tabla 103. Almacén PERTENECE	246
Tabla 104. Almacén PROCESO	246
Tabla 105. Almacén PROCESA.....	246
Tabla 106. Almacén EVALUACION.....	247
Tabla 107. Almacén EVALUA.....	247
Tabla 108. Almacén DOMINIO	247
Tabla 109. Almacén DOMINA	247
Tabla 110. Almacén MODELO_DE_MADUREZ	248
Tabla 111. Almacén SE_RELACIONA.....	248
Tabla 112. Almacén OBJETIVO_DE_CONTROL.....	248
Tabla 113. Almacén CONTIENE	249
Tabla 114. Almacén ACCION	249
Tabla 115. Almacén TIENE_ASOCIADA.....	249
Tabla 116. Almacén METRICA.....	249
Tabla 117. Almacén META	250
Tabla 118. Almacén MIDE.....	250
Tabla 119. Almacén SE_EVALUA	250
Tabla 120. Almacén OBJETIVO_TI	250
Tabla 121. Almacén OBJETIVO_DE_NEGOCIO	251
Tabla 122. Almacén SE_ASOCIA.....	251
Tabla 123. Almacén SE_ASOCIADO	251
Tabla 124. Almacén CONTRASEÑA_ADMINISTRADOR	251
Tabla 125. Almacén CONFIGURACION_BASE_DE_DATOS	252
Tabla 126. Almacén PARAMETROS_INTERFAZ	252
Tabla 127. Entradas Externas de Usuario	253
Tabla 128. Salidas Externas	253
Tabla 129. Consultas Externas	254
Tabla 130. Cálculos Puntos de Función y SLOC.....	254

1. Introducción

1.1 Descripción breve del proyecto

Este proyecto trata del análisis, diseño e implementación de una herramienta software destinada a la resolución de problemas relacionados con las Tecnologías de la Información (TI). La finalidad es proporcionar la solución adecuada a problemas que plantean los objetivos TI en las organizaciones (ya sean pequeñas, medianas o grandes empresas) que quieren utilizar o ya usan metodologías de gobierno de TI.

Existen gran variedad de problemas que surgen a la hora del uso de las TI. Es por ello que se ha hecho necesario un mayor control en el gobierno (o gobernanza) de las TI, es decir, una normalización, método o ajuste que permita de una manera óptima la utilización de las TI con la menor incidencia negativa posible ajustándose a las necesidades empresariales. Esto es lo que se conoce como una correcta alineación en el uso de las TI con los objetivos de una empresa. Esta gestión para una buena administración (buenas prácticas) en el entorno TI ha dado lugar a varias metodologías que definen directrices para llevar una buena gerencia de las TI.

Las principales metodologías existentes relacionadas con el gobierno TI son COBIT, ITIL, CMMI o MOF entre otras. El sistema que se va a implementar está basado en COBIT (y más concretamente en su última versión disponible COBIT 4.1), pues es una metodología, usada internacionalmente, que aglutina de una manera más genérica el mayor número de áreas TI y que además es compatible con el uso al mismo tiempo de otras metodologías como ITIL o CMMI.

Los problemas que surgen en las organizaciones cuando se basan en el gobierno TI pueden ser de muy diversos tipos, existiendo una gran variedad, es por ello que es complicado para una organización identificarlos y dar con una solución factible con la multitud de metodologías existentes hoy día.

La herramienta o sistema propuesto que se va a implementar es capaz de proporcionar una solución o al menos una ayuda o guía para los problemas empresariales que plantean las TI de acuerdo con las metodologías existentes en gobierno TI, en este caso con el uso de COBIT 4.1, otorgando de esta forma una mejor práctica en la alineación del negocio con las TI.

Dicha aplicación consta de dos perfiles de uso: usuario y administrador. El perfil usuario es el que va a utilizar la herramienta para la búsqueda de soluciones para un problema TI.

A partir de un problema TI, dado por el usuario, la aplicación puede identificar dicho problema y dar una solución descrita a través de COBIT. La solución vendría dada

por una serie de acciones, objetivos, metas y métricas que podrían ayudar a la resolución del problema.

Por otro lado, se permiten realizar evaluaciones (*assessments*) de procesos de COBIT, para conocer la madurez de la empresa y cómo podría mejorarse, o ayudar a identificar carencias que puedan repercutir en posteriores problemas de la empresa en el ámbito TI.

Y además de esto, si se trata de un usuario con conocimientos en metodologías TI, puede consultar por las áreas de COBIT, dominio y proceso, para obtener una solución al problema que el usuario tenga en las distintas áreas mencionadas anteriormente.

Es decir, el usuario cuenta con tres ramas distintas en la aplicación para consultar su problema: la primera rama es en la que el usuario escribe su problema, la segunda es en la que el usuario responde unos cuestionarios y la tercera rama es en la que el usuario elige el dominio y proceso que crea que están relacionados con su problema.

Por todas las distintas ramas se llega a un punto en común, que es la solución del problema. Pero antes de mostrar la solución final, la aplicación mostrará los distintos problemas que se han encontrado en la base de datos relacionados con el problema del usuario, el cual, deberá elegir teniendo en cuenta el que más se aproxime a su problema para obtener la solución.

La solución está formada por unas acciones que el usuario debería realizar para solucionar su problema y a modo informativo, se indicarán los objetivos de control, objetivos TI, objetivos de negocio, metas y métricas que COBIT pone a disposición para llegar a realizar una mejor práctica TI-negocio.

El perfil administrador, por otra parte, es capaz de actualizar la herramienta de acuerdo a las necesidades del momento. Se encarga de las opciones típicas de la administración como son las altas, bajas y modificaciones. Por tanto permite dar de alta nuevos problemas, nuevas evaluaciones, nueva información de COBIT, y su borrado y modificación de datos.

Esto permite que en un futuro se puedan incluir nuevos problemas que surjan en el ámbito TI y a su vez nuevas soluciones para resolverlos. Cuantos más problemas existan en la base de datos, más útil será la aplicación a la hora de identificar un mayor número de problemas y sus soluciones. Lógicamente se pueden borrar y modificar dichos datos.

La administración también permite introducir nuevas evaluaciones (tests) borrarlas y modificarlas, así como los modelos de madurez existentes en COBIT relacionados con los diversos tests.

Y por último la aplicación permite que se puedan dar de alta, dar de baja y modificar la información de COBIT lo que supone la posibilidad de actualizar COBIT ante nuevas versiones que puedan surgir, es decir que se permite actualizar la solución a los problemas e información complementaria de la solución dada por COBIT.

Por otra parte, se permite administrar la contraseña que se necesita para entrar en la rama del administrador (la rama del usuario no tiene contraseña ya que se usa

únicamente como consulta). Y también se pueden modificar los valores para conectarse correctamente a la base de datos con la cual el programa se comunica.

Aparte de la administración se cuenta con la opción de modificar distintos parámetros del programa para que la aplicación pueda adaptarse al gusto del usuario. Por ejemplo se puede cambiar el número de preguntas que se muestran en la rama de las evaluaciones, modificar el número de problemas que el usuario puede elegir en la solución de las tres ramas del usuario, se puede cambiar el tamaño de letra de la barra menú del programa así como cambiar el color de la barra de menú, se permite elegir entre idioma inglés o español (versión de prueba en español) y se puede efectuar un cambio a los valores por defecto del programa.

El sistema es una aplicación de interfaz gráfica, es decir una aplicación con ventanas (Java swing) desarrollada en lenguaje Java y cuenta con una base de datos creada por medio del lenguaje SQL implantada en Oracle.

La aplicación se ha creado en inglés puesto que existe una mayor información sobre gobierno TI en este idioma, y en vistas a una posible internacionalización de la aplicación.

Por último decir que para la realización del ciclo de vida del presente proyecto se ha utilizado el modelo iterativo incremental, el cual está basado en varios modelos de cascada realimentados repetidamente de forma iterativa. La idea es realizar las diferentes tareas típicas del ciclo de vida como son el análisis, el diseño, la implementación, la integración y pruebas, y la operación y mantenimiento, en una fase inicial en la primera iteración. A partir de esta realizar las mismas tareas en una segunda iteración con mayor depuración en ellas, a partir de esta iteración otra tercera, esto serían versiones intermedias y así continuar hasta llegar a la versión final.

Mediante este modelo se pueden ir mostrando poco a poco los avances en la aplicación y a su vez poder hacer uso de reutilización y tener la posibilidad de realizar cambios sin mucho coste por la funcionalidad que da el ir incrementando el proyecto fase a fase.

1.2 Objetivos del proyecto

El objetivo fundamental del proyecto es el desarrollo de una aplicación software que aporte soluciones a los objetivos de una empresa en base a conceptos del gobierno TI. La finalidad es poder servir soluciones o ayudas en el uso de las TI a las empresas que las utilizan para que puedan mejorar en sus objetivos y hacerlas más competitivas y con un nivel de madurez mayor, todo ello enfocado hacia el alineamiento de la tecnología con la empresa de acuerdo con el marco COBIT 4.1.

Pero para llevar a cabo este objetivo ha sido necesario alcanzar otros objetivos como han sido:

El estudio y la familiarización de los más conocidos y usados internacionalmente marcos, estándares y metodologías de trabajo y normativas basados en las mejores prácticas tales como COBIT, ITIL, CMMI, MOF, ISO 9001, ISO 27001, ISO 20000...

El aprendizaje del lenguaje de programación Java, especialmente centrado en el diseño de aplicaciones con ventanas en entorno gráfico (tipo Windows) conocido también como Java swing, y el manejo de base de datos por medio del lenguaje SQL de Oracle y su conexión con Java (JDBC).

Y en definitiva, la puesta en práctica de los conocimientos adquiridos durante la carrera, en un proyecto que aglutina diversas áreas de las Tecnologías de la Información como son la Ingeniería del Software, las Bases de Datos, la Programación, y el Gobierno de las Tecnologías de la Información. Todo ello siguiendo el ciclo de vida del software del proyecto con su análisis, diseño, implementación y estudio económicos realizados.

2. Estado del Arte

2.1 ¿Qué se conoce como Gobierno TI?

A menudo se dice que vivimos en la “Era de la Información”. Pero, ¿con qué se asocia normalmente la información en la actualidad?, ¿qué abarca su significado? ¿En qué consiste la Sociedad de la Información o del Conocimiento?

La información es entendida como la adquisición del conocimiento y la comunicación que permiten ampliarla o precisarla sobre cualquier materia o campo. Los datos registrados, de cualquier tipo que sean, son información. Pero el concepto información abarca un significado mayor cuando hablamos de la unión de la información con la tecnología que la gestiona.

El tratamiento que de la información se ha hecho a lo largo de la historia ha alcanzado actualmente el mayor cambio hasta el momento. La manera en que la información es manejada hoy día viene ligada a la revolución que las Tecnologías de la Información o TI (en inglés *Information Technology, IT*) han experimentado a finales del siglo XX y que continúa hasta nuestros días. El avance de las TI ha causado un impacto en la sociedad actual y ha cambiado y ampliado el modo o maneras de hacer negocios, de realizar procesos industriales o empresariales, de comunicación entre la gente, de trabajar, e incluso del uso que del ocio hacemos. Ha pasado a formar parte de todos sin apenas darnos cuenta, llegando a casi todos los aspectos de nuestra vida y es difícil concebirla sin las numerosas tecnologías que se encuentran a nuestra disposición.

La naturalidad con que usamos las nuevas tecnologías actualmente lleva a pensar en cómo sería la vida si no hubieran estado ahí, pues se ha creado una fuerte dependencia de su uso y forman ya parte natural en la sociedad. Pero por supuesto esto no siempre fue así. Desde los comienzos del ser humano, la información ha ido pasando normalmente de generación en generación a través del *boca a boca*, con el consiguiente riesgo de que la información se perdiera para siempre. El ser humano ha tenido la necesidad de comunicarse con sus semejantes y por ello había que encontrar un medio en que la información pudiera perdurar para siempre o lo máximo posible, es decir había que registrar la información de alguna manera. Así se pasó, por poner un ejemplo, desde las prehistóricas pinturas rupestres a la escritura en papiro, o el tallado en madera, piedra o arcilla, hasta la revolución que supuso la invención de la Imprenta. Así llegamos hasta hoy donde está comenzando un nuevo soporte de información, el libro digital (*e-book*). Si bien el *e-book* ya existía por medio de los ordenadores y difundido gracias a Internet, el soporte para el libro digital supone un paso más en el tratamiento de la información y quien sabe si en un futuro sustituirá a los libros tradicionales.

Muchas tecnologías se encuentran en la actualidad a nuestro alcance. Desde los ordenadores, portátiles, Internet, que ha sido una verdadera revolución en sí misma, pasando por móviles, que ya no son meros sistemas de comunicación sino que se tratan de pequeños ordenadores capaces de realizar muchas más funciones como tomar fotos digitales, escuchar música, ver la televisión, texto... , *PDA*s, cámaras digitales, videoconsolas, reproductores y grabadores multimedia, pantallas digitales, micro cadenas de música, televisores, *webcams*, sistemas de almacenamiento, discos duros, *pen-drives*, impresoras, escáneres, periféricos en general, y algunas tecnologías más propias de sectores de investigación, industrial o empresarial como telescopios, microscopios, grúas, servidores, *RAIDs*, nuevos lenguajes de programación orientados a diversos campos y con características propias para nuevas tecnologías, maquinaria industrial controlada informáticamente en muy diversos sectores, en el automovilístico, sanitario, e incluso en las pequeña y mediana empresa... todas estas tecnologías y otras muchas usadas hoy día entrarían dentro de las tecnologías conocidas como Tecnologías de la Información puesto que tratan y gestionan todo tipo de datos. Así pues cuando hablamos de la Información es muy probable que estemos hablando de Tecnologías de la Información, es difícil separar un concepto del otro ya que prácticamente van unidos.

Las TI, Tecnologías de la Información, es también conocida como TIC, Tecnologías de la Información o Sistemas de la Información TSI y Comunicaciones. Es un concepto muy amplio que abarca multitud de ámbitos concernientes a la administración y procesamiento de la información. Además, como la investigación está en constante cambio, su significado puede variar a lo largo del tiempo, porque puede cambiar el modo en que los datos o la información se manejan. Es por eso que hay muy diversos significados que intentan explicar de qué tratan las TI, pero no son significados excluyentes sino que cada uno de ellos aporta un punto de vista que enriquece al significado de las TI. Una posible definición sería que la TI es el estudio, diseño, desarrollo, implementación, soporte o dirección de los sistemas de información en general computarizados, asociados al sector tecnológico de informática computacional y comunicaciones, en particular de software de aplicación y hardware de computadores. En términos más generales, la TI sería cualquier tecnología que trata la producción, manipulación, almacenamiento, comunicación y difusión de la información.

Los profesionales TI realizan una variedad de tareas entre las que encontramos el diseño, el desarrollo, soporte y administración software, hardware y redes informáticas. Desde instalación de simples aplicaciones software como redactar contratos, bases de datos de clientes, etc., a aplicaciones más complejas como animación en videojuegos o películas, comercio a través de Internet, a diseño de complejas redes de computación y bases de datos como mantenimiento de satélites espaciales, ingeniería de software y hardware para gestión de sistemas completos de información.

La revolución tecnológica que vivimos y que ha impactado en el sector de la industria informática ha producido como consecuencia importantes cambios en el resto de los sectores de la sociedad. Cada vez un mayor número de organizaciones considera que la información y la tecnología asociada a ella representan sus activos más importantes. Las empresas y organizaciones dependen de los órdenes económicos industriales, y sociales en los que se encuentran inmersas por lo que, si las tendencias tecnológicas y los entornos económicos e industriales cambian, deben adaptarse rápidamente a las nuevas circunstancias para sobrevivir. Por tanto la gerencia de una organización debe establecer un sistema de control adecuado para soportar debidamente los procesos del negocio.

Lógicamente, las innovaciones en cualquier campo crean oportunidades pero provocan también nuevos riesgos a veces desconocidos por lo que se hace necesario e indispensable llevar un control de todo lo que se realiza, de cómo debe realizarse y de lo que se quiere conseguir y así poder llevar un control adecuado para los intereses de una organización.

Las actividades de TI son esenciales para gestionar las transacciones, la información y los conocimientos necesarios para iniciar y sostener su economía, además de mantener los objetivos, el prestigio y la continuidad del negocio. Las entidades dependen cada vez más de estas actividades para tener éxito pues es fundamental para apoyar, mantener y hacer crecer el negocio. Además algunas legislaciones sobre TI que han surgido en diferentes países han ayudado a la consolidación de los modelos que controlan las TI, leyes como la LOPD (Ley Orgánica de Protección de Datos) y LSSI (Ley de la Sociedad de los Servicios de Información) en España o la ley Sarbanes-Oxley (SOX) en Estados Unidos que se centra en mejorar el gobierno corporativo por medio de medidas que aumentan el control interno y refuerzan la responsabilidad corporativa, como respuesta a diversos escándalos financieros.

Además de esto, los usuarios buscan confiar en los sistemas que utilizan. Confianza es la ausencia de sorpresas y la certeza, razonable de que alcanzaremos los objetivos deseados. La confianza puede tener infinitos orígenes, pero no hay confianza que dure si no hay seguridad. Y la seguridad tiene algunas características esenciales que todo sistema de información debería cumplir para obtener un equilibrio entre el riesgo y las medidas establecidas para paliarlo:

Confidencialidad: La información solo debe ser accesible a las personas autorizadas.

Integridad: La información debe mantenerse tal y como se creó, exacta y completa en todo momento, solo modificada por personal habilitado.

Disponibilidad: Asegurar que cuando queramos acceder a la información se pueda, es decir, utilizable cuando y como lo requieran los usuarios autorizados.

Autenticidad: Asegurar el origen y destino de la información por medio de firma electrónica, validez de *email*... Asegurar la autoría del documento.

Imposibilidad de Rechazo (No repudio): Asegurar que cualquier entidad que envíe o recibe información, no alegue que no lo hizo.

El fin de todo ello es proteger los bienes que pueden ser hardware, software, datos, procesos telemáticos, es decir los sistemas de información que manejan las TI, de las amenazas que se puedan provocar en las vulnerabilidades del sistema a través de métricas para solucionar los problemas.

Es por ello que hoy día se hace casi obligado llevar un control de las Tecnologías de la Información que se usan en multitud de empresas ya sea para iniciar su actividad o mejorarla. Todo este control de las TI es conocido como Gobierno de las Tecnologías de la Información.

El concepto de gobierno o gobernanza de las Tecnologías de la Información tiene sus orígenes en los años 80, aunque toma fuerza como tal en la primera década del año 2000. Desde hace unos años están tomando gran importancia ya no tanto las cuestiones más tecnológicas sino las cuestiones más relacionadas con su gestión y gobierno. Por primera vez adquiere una mayor importancia la administración y la dirección antes que la tecnología. Esto no quiere decir que se abandone el campo técnico o tecnológico, se debe hacer hincapié en la técnica también puesto que muchas actividades están relacionadas con ella. No se deja de ser experto en técnica sino que se es en técnica y dirección, ambos aspectos a la vez.

Existen también varias definiciones en torno al concepto de Gobierno de las TI. Webb et al. (2006) propone la siguiente definición:

“El gobierno de TSI es el alineamiento estratégico de TSI con la organización tal que se consigue el máximo valor de negocio por medio del desarrollo y mantenimiento de un control y responsabilidades efectivas, gestión del desempeño y gestión de riesgos de TSI”.

El IT Governance Institute (ITGI) propone la siguiente definición:

“El gobierno es el conjunto de responsabilidades y prácticas ejercidas por el Consejo de Administración de la empresa y por los principales Directivos con la finalidad de proporcionar una dirección estratégica, asegurando que los objetivos de la empresa son conseguidos, asegurando que los riesgos son gestionados apropiadamente y verificando que los recursos de la empresa son empleados de forma responsable”.

Los principales desafíos a los que las organizaciones se enfrentan hoy en día son los siguientes:

- **Alineamiento estratégico de TI.**

Consiste en asegurar que la estrategia de TI se encuentre alineada con la estrategia del negocio. Es decir, que las TI sean las apropiadas para dar servicio de acuerdo a la estrategia que soporta la empresa para que estén disponibles para prestar los servicios necesarios y requeridos priorizando el uso de los recursos TI y de este modo permitan transformar o crecer la entidad dando un mejor servicio.

- **Entrega de valor TI.**

Se trata de obtener una entrega exitosa de valor de negocio a la organización a través de la TI, asegurando que la TI entregan los beneficios prometidos respecto a la estrategia, concentrándose en la optimización de costes proporcionando el valor intrínseco de la propia TI.

- **Gestión de riesgos.**

Se debe identificar y mitigar los riesgos para preservar el valor. Esto supone un importante desafío para los directores que se enfrentan con la complejidad técnica, dependencia de proveedores de servicio y la información fiable o no sobre monitorización de riesgos. Se destacan los principales riesgos a considerar: inversión o gastos, seguridad y acceso, integridad, relevancia, disponibilidad, infraestructura y responsabilidades de los diferentes roles.

- **Gestión de recursos TI.**

Es preciso establecer y desplegar las capacidades de TI para las necesidades de la organización. Tener en cuenta los retornos de las inversiones esperados, preparación de casos de negocio formalizados y consistentes, niveles de riesgo, aplicar métricas, etc.

- **Medición del rendimiento.**

Verificar en todo momento si es necesario un re direccionamiento del alineamiento con un bucle de realimentación a través de un cuadro de mando integral (*balanced scorecard*) para poder conseguir el alineamiento entre negocio y TI. Además así podremos asegurar un gobierno efectivo de TI y orientar en el futuro para proporcionar mejores servicios y obtener un crecimiento de aprendizaje continuo.

Los objetivos de estos dominios de gobierno de TI serían:

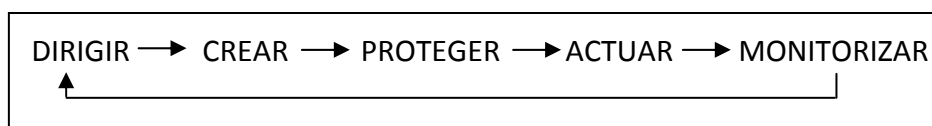


Ilustración 1. Objetivos de dominios de gobierno de TI

Estos problemas deben ser abordados y promovidos a través de medidas eficaces y oportunas de la alta dirección que gestiona una empresa. Es necesario un marco de gestión eficaz de TI que se ocupe del alineamiento estratégico, la gestión del desempeño, la gestión de riesgos, control y responsabilidades, la entrega de valor y gestión de recursos. Este marco de gestión de gobierno de TI y su aplicación efectiva son responsabilidades de la junta directiva y la gestión ejecutiva. El gobierno de TI es una parte integrante del gobierno empresarial que consiste en el liderazgo y las estructuras organizativas y procesos que garantizan que la organización sostiene y amplía las estrategias de la organización y los objetivos.

Debido a la variedad y complejidad de la TI han surgido modelos, algunos de ellos tomados como estándares, que nos indican cómo gestionar eficaz y eficientemente gran variedad de procesos que conciernen al mundo TI y que son de vital importancia para las organizaciones. Pero estos modelos, como en algunos de ellos se sugiere, no deberían ser tomados en todo caso como la verdad absoluta o máxima pues en la práctica hay muchos factores a tener en cuenta y puede que no se deba en algunos aspectos o determinados momentos aplicar a raja tabla la totalidad de procesos o actividades en el negocio. Con el fin de que este prospere, al fin y al cabo es de lo que se trata, del que negocio crezca, no de que se vea mermado por medidas que no se ajustan a la realidad de la empresa, se deben tomar las decisiones que la organización crea oportunas. Si bien hay que tener en cuenta que al principio de las decisiones, aplicando estos modelos, puede suponer una pérdida inicial que tras un período de tiempo puede retornar valor (es lo que se conoce como el retorno de la inversión ROI). Habrá que estudiar el impacto del riesgo para el tipo de organización que se trate.

Todos estos modelos, a pesar de tener sus propias definiciones y esquemas para afrontar los objetivos perseguidos, siguen un mismo patrón de comportamiento:

1. **Definir o dirigir** los fines, finalidad y objetivos del proyecto.
2. **Medir** o identificar las tareas estableciendo líneas base para distintos niveles.
3. **Analizar** o evaluar la información, los datos obtenidos en las monitorizaciones para identificar patrones de comportamiento.
4. **Mejorar**, desarrollar e implementar soluciones paliativas para carencias detectadas.
5. **Controlar** o retroalimentar (feedback) para asegurar que los problemas sean reconocibles y puedan ser solucionados y que los nuevos métodos sean aplicables.

Entre estos modelos o marcos de gobierno de TI, también conocidos como guías de mejores prácticas de gestión de servicios TI, podemos encontrar entre los más reconocidos a: COBIT, ITIL, CMMI, MOF, COSO (centrado en gobierno de una organización en general más que en TI en particular)...

2.2 ¿Qué es COBIT?

COBIT (*Control Objectives for Information and related Technology*). (Objetivos de Control para la Información y las Tecnologías). COBIT fue creado por Information Systems Audit and Control Association (ISACA) e IT Governance Institute (ITGI) que como conjunto de profesionales de diferentes áreas TI realizaron una guía para tratar los problemas sobre el control de las TI. La primera edición de COBIT data de 1996. Otras ediciones posteriores han sido en 1998, 2000 y 2006 (versión 4.0). La última versión de COBIT publicada por ITGI es la edición 4.1 del año 2007.

COBIT es un conjunto estructurado de dominios, procesos y actividades que ofrecen “buenas prácticas” a través de un marco referencial que presenta una estructura manejable y lógica metodológica para ayudar en su aplicación al gobierno TI. Sirve de ayuda para las brechas existentes entre riesgos de negocio, necesidades de control y aspectos técnicos.

Su misión es la de investigar, desarrollar, publicar y promover un conjunto internacional actualizado de objetivos de control para las TI que sea de uso cotidiano para gerentes, auditores y profesionales TI.

La nueva edición enfatiza el cumplimiento de normas y legislación, sin invalidar los conceptos fundamentales de las versiones anteriores. Se incluyen mejoras y avances desde un criterio de practicidad y facilidad, teniendo en cuenta las propias experiencias de profesionales y asociaciones de profesionales que colaboran en su elaboración, de acuerdo a los cambios en las TI y a las experiencias prácticas que cada uno aporta, todo ello dentro del gobierno TI.

COBIT surgió dentro del ámbito de la Auditoría de sistemas y está principalmente orientado al control de las actividades, pero no solo se focaliza en los servicios de explotación sino que cubre todo el espectro relacionado con los sistemas y tecnologías de la información.

COBIT apoya y sustenta el gobierno de las TI, proporcionando un marco de referencia que asegure que:

- Las TI están alineadas con el negocio.
- Las TI posibilitan la realización de la actividad de negocio al mismo tiempo que contribuyen a la maximización de los beneficios.
- Los recursos de las TI (humanos y técnicos) son utilizados de forma responsable.
- Los riesgos de las TI son gestionados y dirigidos adecuadamente.

Estructura de COBIT

COBIT 4.1 proporciona un *framework* (esquema, marco de referencia o estructura...) práctico por medio de procesos y dominios en los que se incluye:

- **Un resumen ejecutivo.**

Es una síntesis dirigida a la alta gerencia que proporciona entendimiento y conciencia sobre los conceptos clave y principios de COBIT, tales como quién debe promover la instauración de la metodología, métricas y controles para llevar a cabo un buen sistema de gobierno TI en las empresas. Presenta los antecedentes y la estructura básica de COBIT describiendo de manera general los procesos, los recursos y los criterios de información que forman la “columna vertebral” de COBIT.

- **El esquema o marco de referencia (*framework*).**

Describe los cuatro dominios y los 34 procesos, características del modelo del gobierno de las TI basado en controles e identifica los requerimientos de negocio para la información y los recursos TI que impactan de forma directa por cada objetivo de control. Presenta las guías de navegación para que los lectores se orienten en la exploración del material de COBIT.

- **El contenido central o *core*.**

Se exponen los objetivos de control para los 34 procesos de cada uno de los 4 dominios de gobierno TI, metas, métricas, actividades, directrices de gestión y modelos de madurez. Integra en el contenido a su vez lo expuesto en el resumen ejecutivo y en el esquema.

- **Apéndices.**

Se trata la relación con otras normas, glosario, diferencias con versiones anteriores de COBIT, guías de auditoría, modelo de madurez para el control interno, factores críticos de éxito, indicadores clave de objetivo e indicadores clave de rendimiento...

La gran ventaja de COBIT 4.1 es su estructura de presentación. Está estructurado con un orden lógico y racional de mayor nivel a menor nivel, que permite una mayor comprensión del esquema, la graduación en su implantación, su revisión y actualización periódica sin tener que cambiar todo el esquema, a la vez que el enfoque de procesos permite una mayor adaptabilidad a los procesos de negocio y operativos de una organización. A la vez permite visualizar y tener una guía para algo tan importante en control interno como es la interrelación entre los objetivos de control de alto nivel y de detalle según su agrupación por actividades y según los requerimientos para estos controles.

COBIT 4.1 se estructura en 34 procesos (también llamados objetivos de control de alto nivel) para los procesos de las TI, que están agrupados en cuatro dominios de actividades típicas del gobierno de las TI. Dentro de cada dominio en que se agrupan los 34 procesos se consideran los riesgos (análisis y evaluación) y el establecimiento de controles.

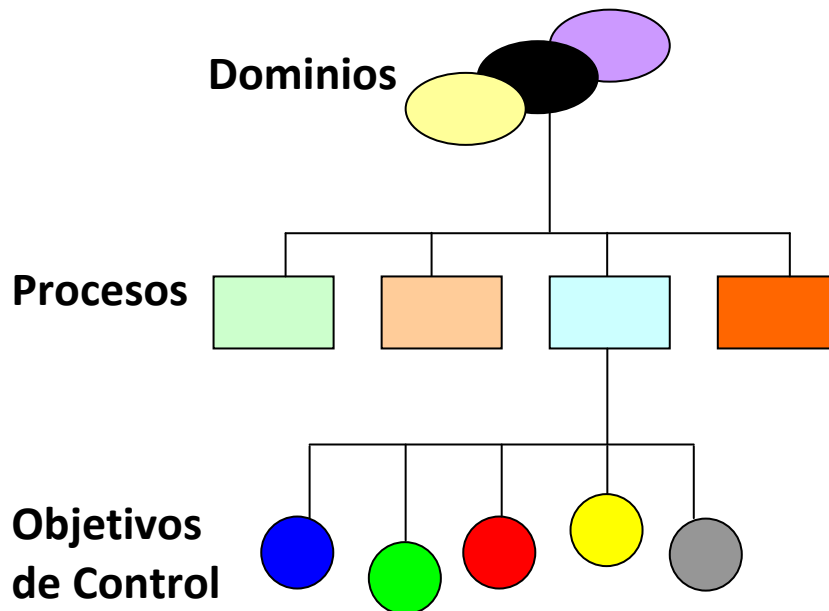


Ilustración 2. Estructura de COBIT

Los 4 dominios de COBIT son:

1. Planificar y Organizar
2. Adquirir e Implantar
3. Entregar y Dar Soporte
4. Monitorizar y Evaluar

Para cada uno de estos 4 dominios existen un total de 34 procesos. Para el dominio 1 hay 10 procesos, para el dominio 2 hay 7 procesos, para el dominio 3 hay 13 y para el dominio 4 existen 4 procesos, sumando así el total de 34 procesos. A su vez, a cada proceso le corresponde un número de subprocesos llamados objetivos de control (en otras terminologías, también conocidos como objetivos de control de nivel de detalle). En total existen 215 objetivos de control.

PLANIFICAR Y ORGANIZAR (PO)

Este dominio cubre la estrategia y las tácticas, y está relacionado con la identificación de las formas en que la tecnología de la información puede contribuir al alcance de los objetivos de negocio. Debe establecerse una organización y una infraestructura tecnológica apropiada. Además la realización de una visión estratégica

requiere que ésta sea planificada, comunicada y gestionada desde diferentes perspectivas. Los 10 procesos que forman este dominio son:

- PO1. Definir el plan estratégico de TI**
- PO2. Definir la estructura de Información**
- PO3. Establecer la Dirección Tecnológica**
- PO4. Definir los procesos de TI, su organización y sus relaciones**
- PO5. Gestionar las inversiones en TI**
- PO6. Comunicar los objetivos y orientación de la Gerencia**
- PO7. Gestionar los Recursos Humanos de TI**
- PO8. Gestionar la Calidad**
- PO9. Evaluar y gestionar los riesgos de TI**
- PO10. Gestionar los Proyectos**

ADQUIRIR E IMPLEMENTAR (AI)

Para llevar a cabo la estrategia de TI, las soluciones de TI deben ser identificadas, desarrolladas o adquiridas, así como implementadas e integradas dentro del proceso del negocio. Además, este dominio cubre los cambios y el mantenimiento realizados a sistemas existentes.

- AI1. Identificar Soluciones Automatizadas**
- AI2. Adquirir y Mantener el Software de Aplicaciones**
- AI3. Adquirir y Mantener la Infraestructura Tecnológica**
- AI4. Posibilitar su Funcionamiento y Uso**
- AI5. Procurar los Recursos de TI**
- AI6. Gestionar los cambios**
- AI7. Instalar y Acreditar las Soluciones y los Cambios**

ENTREGAR Y DAR SOPORTE (DS)

En este dominio se hace referencia a la entrega de los servicios requeridos, que abarca desde las operaciones tradicionales hasta el entrenamiento, pasando por seguridad y aspectos de continuidad. Con el fin de proveer servicios, deberán establecerse los procesos de soporte necesarios. Este dominio incluye el procesamiento de los datos por sistemas de aplicación, frecuentemente clasificados como controles de aplicación.

- DS1. Definir y Gestionar los Niveles de Servicio**
- DS2. Gestionar los Servicios de Terceros**
- DS3. Gestionar el Rendimiento y la Capacidad**

- DS4. Garantizar la Continuidad del Servicio**
- DS5. Garantizar la Seguridad de los Sistemas**
- DS6. Identificar y Asignar los Costes**
- DS7. Concienciar y Formar a los Usuarios**
- DS8. Gestionar la Asistencia a Usuarios y los Incidentes**
- DS9. Gestionar la Configuración**
- DS10. Gestionar los Problemas**
- DS11. Gestionar la Información**
- DS12. Gestionar el Entorno físico**
- DS13. Gestionar las Operaciones**

MONITORIZAR Y EVALUAR (ME)

Todos los procesos necesitan ser evaluados regularmente a través del tiempo para verificar su calidad y suficiencia en cuanto a los requerimientos de control, integridad y confidencialidad. Este es, precisamente, el ámbito de este dominio.

- ME1. Monitorizar y Evaluar el Rendimiento de TI**
- ME2. Monitorizar y Evaluar el Control Interno**
- ME3. Garantizar el Cumplimiento Legal y Reglamentario**
- ME4. Proporcionar Gobierno para TI**

Todos los procesos y objetivos de control necesitan además de otros elementos a considerar para implantar adecuadamente un gobierno TI. Estos elementos están relacionados con los requerimientos del negocio con respecto a los servicios TI y con los recursos que componen las TI. Por cada proceso, COBIT incluye los siguientes aspectos a tener en cuenta:

1. Requerimientos del Negocio con respecto a las TI:

- Efectividad o eficacia
- Eficiencia
- Confidencialidad
- Integridad
- Disponibilidad
- Cumplimiento
- Confiabilidad

En el apartado respectivo se dan indicaciones de si el objetivo de control que se trata está relacionado de forma Principal o Secundaria con cada uno de los requerimientos. En algunos casos pueden estar todos relacionados o solo alguno de ellos.

2. Recursos de TI afectados:

- Aplicaciones
- Información
- Infraestructura
- Personal

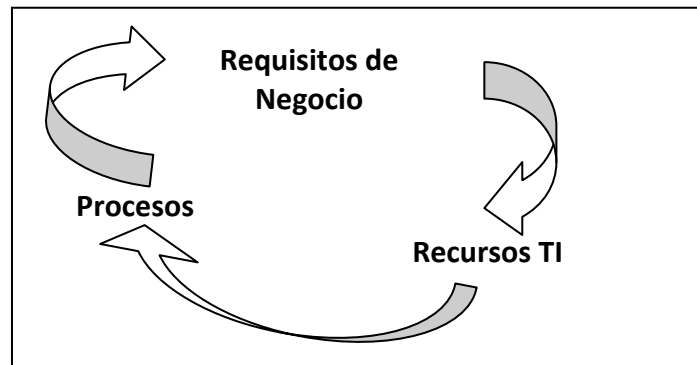


Ilustración 3. Evaluación del Riesgo

3. Áreas Centrales para el Gobierno de TI:

- Alineación Estratégica
- Entrega y Servicio que añade valor
- Gestión de los Recursos
- Gestión del Riesgo
- Medición del Rendimiento

4. Objetivos de Control:

Se enumeran los objetivos de control con explicaciones sobre su propósito y alcance.

5. Directrices de Gestión:

Se dispone de sugerencias y guías sobre las interrelaciones con otros dominios y procesos. Las sugerencias señalan la relación de recepción (input) con otros objetivos de control e inclusive externos a COBIT y la entrega hacia otros objetivos (output).

6. Responsabilidades por los distintos niveles de Dirección y Gerencia:

Las responsabilidades se incluyen tanto desde la Alta Gerencia, como a unidades de negocio, hasta Auditoría.

La definición y el alcance de las responsabilidades provienen de un cuadro de responsabilidades denominado RACI (Responsible, Accountable, Consulted and Informed).

7. Cuadro de objetivos y métricas aplicables:

Se muestran los objetivos y métrica aplicables para el objetivo de control tratado y los indicadores para la respectiva medición.

- Objetivos de la actividad
- Objetivos de los procesos
- Objetivos de TI:
 - Indicadores clave de rendimiento
 - Indicadores de los objetivos clave de los procesos
 - Indicadores de los objetivos clave de TI

8. Modelo de Madurez:

Se indican los criterios para considerar cuando se está en la situación de:

- 0 – No existente
- 1 – Inicial
- 2 – Repetible
- 3 – Definido
- 4 – Gestionado
- 5 – Optimizado

COBIT está reconocido en todo el mundo y ha sido implantado en numerosas instituciones, muchas de ellas internacionales en Estados Unidos, Australia, Europa, África, América Latina y Asia.

COBIT abarca todos los aspectos de las TI, y no un solo aspecto. Es por tanto la cobertura mayor y se puede implantar sin problema junto a otras normas que lo complementen. Ayuda al gobierno TI en su globalidad e integralmente pero no es una norma de seguridad de la información y los sistemas, para ello existen normas complementarias que funcionan correctamente si son usadas junto a COBIT. Deben ser usadas unas normas u otras dependiendo de las necesidades específicas de una organización. COBIT puede funcionar correctamente junto a otras normas y estándares como ITIL, CMMI y algunas ISO.

La implantación de COBIT debe realizarse teniendo en cuenta que el esquema aplicado sea eficiente, adaptable en el tiempo a los cambios que surjan, no burocrático y que aporte valor a la gestión de las TI. Debe evitarse que se dificulte el desarrollo de

las actividades por un seguimiento demasiado rígido del modelo que implique unos gastos elevados u otras pérdidas.

2.3 ¿Qué es ITIL?

ITIL (*Information Technology Infrastructure Library*). (Biblioteca de Infraestructura de las Tecnologías de la Información TSI). ITIL proporciona una metodología que permite ir implantando en la empresa un conjunto de prácticas y procesos, posibilitando la mejora de calidad de los servicios ofrecidos, y la medición de la misma. Ofrece, en definitiva, una capacidad de gestión y mejora continua.

ITIL fue desarrollada por la Agencia Central de Computación y Telecomunicaciones (Central Computer and Telecommunications Agency – CCTA) de Gran Bretaña a finales de la década de los 80. No obstante, es hacia mediados de los años 90 cuando se empieza a convertir en un estándar de ámbito mundial para la Gestión de Servicios de una manera eficiente y consistente. En abril de 2001, la OGC (Office of Government Commerce) absorbió a la CCTA y actualmente la OGC es propietaria de ITIL.

ITIL se hace popular debido a que presenta un marco de trabajo no propietario, y que se puede aplicar en todo tipo de organizaciones, ya sean pequeñas, medianas, y/o grandes. ITIL ofrece un conjunto de mejores prácticas de dirección y gestión de servicios a implantar en una organización con el objeto de mejorar notablemente la calidad de los servicios proporcionados. Estas prácticas, indican qué se debe hacer pero no exactamente cómo.

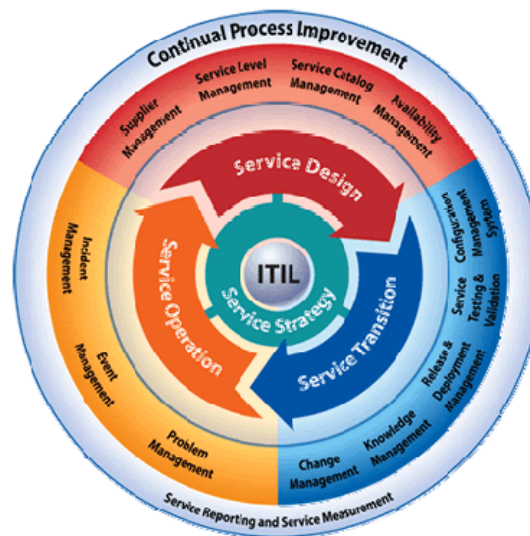


Ilustración 4. Ciclo de Vida de ITIL

ITIL es un conjunto de libros publicados por la OGC, que permiten mejorar la calidad de los servicios de TSI que presta una organización a sus clientes.

Hay que diferenciar dos versiones de ITIL, las cuales se describen a continuación:

ITIL Versión 2 (ITIL V2)

Las publicaciones de ITIL V2 son las siguientes:

- Gestión de la Infraestructura TI (ICT Infrastructure Management, 2002)
- Gestión de Aplicaciones (Applications Management, 2002)
- Planificación para la Aplicación de los Servicios de Gestión (Planning to Implement Service Management, 2002)
- Perspectiva de Negocio (The Business Perspective, 2005)
- Soporte de Servicio (Service Support, 2000)
- Provisión de Servicio (Service Delivery, 2001)
- Gestión de Seguridad (Security Management, 1999)

Cada uno de los libros cubre un área abarcada por ITIL y se desarrolla en un documento distinto publicado por la OGC.

Descripción de las publicaciones y sus procesos principales:

- **Soporte de Servicio:** Este proceso de ITIL se centra en asegurar que el cliente, ya sea de tipo interno o externo, recibe el servicio adecuadamente, y que éste se gestiona internamente de la mejor forma posible. Cubre los siguientes procesos:
 1. Centro de Servicios
 2. Gestión de Incidentes
 3. Gestión de Problemas
 4. Gestión de Configuración
 5. Gestión de Cambios
 6. Gestión de la Entrega
- **Provisión de Servicio:** Este proceso de ITIL trata todos los aspectos de la administración de los servicios de soporte y mantenimiento que se prestan al cliente. La Provisión de Servicio, en conjunto con el Soporte de Servicio, son el pilar más importante del marco de trabajo de ITIL para la Gestión de Servicios TSI. La Provisión de Servicio está compuesta por los siguientes procesos:
 1. Gestión de Niveles de Servicio
 2. Gestión Financiera de los Servicios TSI
 3. Gestión de la Capacidad
 4. Gestión de la Continuidad de los Servicios TSI
 5. Gestión de la Disponibilidad
- **Gestión de la Seguridad:** Este proceso de ITIL tiene como punto de inicio los procesos existentes de ITIL y les añade las actividades de Gestión de

Seguridad. El proceso de Gestión de Seguridad se encarga de gestionar un nivel de seguridad definido para la información y los servicios de TSI. Este proceso forma parte del trabajo de todos los directores, por lo tanto, es la Dirección la responsable de tomar medidas adecuadas para reducir las posibilidades de fallo hasta un nivel que se considere aceptable.

- **Perspectiva de Negocio:** La Perspectiva de Negocio, tiene como objetivo la familiarización del negocio con los componentes que sirven de base de la infraestructura de Tecnología de la Información y las Comunicaciones (TIC). La Perspectiva de Negocio ayuda al negocio a comprender los beneficios de las Mejores Prácticas en la Gestión del Servicio de TSI, de manera que puedan ser mejor informados y más capaces de gestionar las relaciones con los proveedores del servicio para obtener un beneficio mutuo. También ayuda al Proveedor del Servicio a utilizar términos llanos con el negocio.
- **Planificación para la Implementación de los Servicios de Gestión:** Este libro examina los problemas y las tareas implicadas al planificar, implementar y mejorar los procesos de Gestión de Servicio dentro de una organización. Se centra en aspectos clave a tener en cuenta cuando se planifica la implementación de la Gestión de Servicio de TSI.

El principal objetivo de la Gestión de Servicio es asegurar que todos los servicios de TSI están alineados con las necesidades del negocio. La mayoría de las organizaciones que utilizan TSI serán dependientes de ella. Si los procesos de TSI no están implementados, gestionados y soportados en el lugar adecuado, el negocio probablemente sufrirá una degradación considerable en términos de pérdida de horas productivas, costes más altos, así como por ejemplo una pérdida de ingresos o incluso la quiebra del negocio.

La Provisión de Servicio de TSI en todas las organizaciones necesita compararse con las demandas actuales y cambiantes rápidamente del negocio.

- **Gestión de Aplicaciones:** La Gestión de Aplicaciones trata el aspecto complejo de gestionar las aplicaciones desde la necesidad inicial del negocio, a través del ciclo de vida de la Gestión de la Aplicación, hasta la retirada. También, la Gestión de Aplicaciones incluye la interacción con las disciplinas de Gestión de Servicio de TSI incluidas en Provisión de Servicio, Soporte de Servicio y Gestión de la Infraestructura de TIC. Se hace énfasis en asegurar que los proyectos y estrategias de TSI estén fuertemente alineadas con las del negocio a través del ciclo de vida de las aplicaciones.

- **Gestión de la Infraestructura TSI:** Hoy en día, las infraestructuras de TSI son la base sobre la que se construyen los servicios del negocio. La Infraestructura TSI está compuesta por los siguientes procesos:
 1. **Diseño y Planificación:** Tiene como objetivo la creación y/o mejora de la solución TSI.
 2. **Despliegue:** Tiene como objetivo la implementación y la extensión del negocio y/o solución de TSI según se diseñó y planificó, con una mínima interrupción en los procesos del negocio.
 3. **Operaciones:** Este proceso tiene como objetivo mantener diariamente la infraestructura de TSI.
 4. **Soporte Técnico:** Su objetivo es garantizar los servicios entregados por la Gestión de la Infraestructura de TSI, interesándose en la estructura y apuntalamiento de otros procesos.

ITIL Versión 3 (ITIL V3)

En esta nueva versión de ITIL, se re estructura el Ciclo de Vida del Servicio.

La principal diferencia entre ITIL V3 e ITIL V2 radica en la nueva estructura del Ciclo de Vida del Servicio de la versión 3, orientada a una oferta de servicios de TI más exitosa a largo plazo para el cliente.

Esta reestructuración tiene como objetivo proporcionar un enfoque empresarial para grandes corporaciones que utilizan de forma extensa ITIL en sus operaciones, y además consolidar el modelo de cara a conseguir mejores resultados. Es por ello, que no se recomienda utilizar ITIL V3 en empresas emergentes o medianas si no cuentan ya con un modelo ITIL bien consolidado y no se aspira a una expansión de la empresa en un corto período.

Las publicaciones de ITIL V3, basadas en el ciclo de vida del servicio, son las siguientes:

- **Estrategia del Servicio**
- **Diseño del Servicio**
- **Transición del Servicio**
- **Operación del Servicio**
- **Mejora Continua del Servicio**

Descripción de las publicaciones y sus procesos principales:

- **Estrategia del Servicio:** Se estudia el mercado y se valoran las posibilidades de la prestación de nuevos servicios que sean innovadores y que satisfagan al cliente. Además, se analizan los servicios existentes de cara a implantar posibles mejoras en los mismos. También se verifican los contratos con los proveedores, teniendo en cuenta a posibles nuevos proveedores para la empresa. La Estrategia del Servicio está compuesta por los siguientes procesos:

1. Gestión Financiera
2. Gestión del Portafolio
3. Gestión de la Demanda

- **Diseño del Servicio:** Una vez se ha identificado un posible servicio, hay que analizar si es viable o no. Y de esto es de lo que se encarga esta publicación. Para ello se tienen en cuenta factores como la infraestructura de que dispone la empresa, la capacitación del personal, y se planifican aspectos clave como la seguridad y la prevención ante desastres de varios tipos. En caso de que su puesta en marcha se llevara a cabo, se tiene en consideración la reasignación de cargos, y la

infraestructura y el software a implementar. El Diseño del Servicio está compuesto por los siguientes procesos:

1. Gestión del Catálogo de Servicios
2. Gestión de Niveles de Servicio
3. Gestión de la Disponibilidad
4. Gestión de la Capacidad
5. Gestión de la Continuidad de los Servicios de TI
6. Gestión de Proveedores
7. Gestión de la Seguridad de Información

- **Transición del Servicio:** Una vez se da el visto bueno al servicio, antes de ponerlo en marcha se debe realizar pruebas. Para ello, se analiza la información disponible acerca del nivel real de capacitación de los posibles usuarios, o los recursos IT disponibles, entre otros aspectos a tener en cuenta. Después de este análisis, se prepara un escenario real para realizar las pruebas, y tras finalizarlas se destruye. Por último, se evalúan los resultados obtenidos, haciendo una comparación entre las expectativas esperadas con los resultados conseguidos durante las pruebas. La Transición del Servicio está compuesta por los siguientes procesos:

1. Gestión de la Configuración y Activos
2. Gestión del Cambio
3. Gestión del Conocimiento
4. Planificación y Apoyo a la Transición
5. Gestión de Release y Despliegue
6. Gestión Validación y Pruebas
7. Evaluación

- **Operación del Servicio:** Una vez puesto en marcha el servicio, se monitoriza y controla el mismo: eventos, incidencias, problemas, peticiones, y accesos al servicio. La Operación del Servicio está compuesta por los siguientes procesos:

1. Gestión de Incidentes
2. Gestión de Problemas
3. Cumplimiento de Solicitudes
4. Gestión de Eventos
5. Gestión de Accesos

- **Mejora Continua del Servicio:** Siempre con la visión de proporcionar la máxima calidad posible del servicio, se utilizan herramientas de medición y feedback para realizar informes acerca de su funcionamiento. Para ello, se debe verificar el nivel de conocimiento de los usuarios respecto al nuevo servicio implantado, fomentar el registro

e investigación referentes al servicio, y dispone de la información del resto de usuarios.

2.4 ¿Qué es CMMI?

CMMI (*Capability Maturity Model Integration*). (Modelo de Integración y Capacidad de Madurez). CMMI es una metodología encargada de definir y gestionar los procesos a realizar por una organización. Fue desarrollado inicialmente para los procesos relativos al software por la Universidad Carnegie-Mellon para el Instituto de Ingeniería del Software SEI (Software Engineering Institute) de los Estados Unidos de América. En diciembre del año 2000, el SEI publicó un nuevo modelo, el CMM

Básicamente, CMMI proporciona un conjunto de normas para calidad enfocadas al mundo del software, que se aplican a los diferentes procesos que hay que llevar a cabo para producir software con calidad. Es importante señalar que este modelo nos dice qué hay que hacer, pero no cómo hay que hacerlo.

Entre sus principales características principales se encuentran:

- Disminuir o eliminar la redundancia en el trabajo
- Aumentar la fiabilidad en la predicción de costos
- Aumenta el rehúso de productos y procesos
- Disminuir costos debido a través de la evaluación y mejora continua de los procesos

CMMI incluye las herramientas para la implementación de:

Ingeniería de Sistemas (Systems Engineering SE), Ingeniería de Software (Software Engineering SW), Desarrollo Integrado del Producto y Procesos (Integrated Product and Process Development IPPD), y Suplidor de Recursos (Supply Sourcing SS).

Madurez en CMMI

CMMI muestra un modelo basado en 5 niveles de madurez, en función de los procesos de calidad implantados en los procesos de la empresa.



Ilustración 5. Nivel 1 o Nivel Inicial de Madurez

Nivel 1 o Nivel Inicial de Madurez:

Estado inicial donde el desarrollo se basa en la heroicidad y responsabilidad de los individuos.

- Los procedimientos son inexistentes o localizados a áreas concretas
- No existen plantillas definidas a nivel corporativo

Nivel 2 o Nivel Administrado:

Se normalizan las buenas prácticas en el desarrollo de proyectos (en base a la experiencia y al método).

- En este nivel las buenas prácticas se mantienen en los momentos de estrés
- Están definidos los productos a realizar
- Se definen hitos para la revisión de los productos

Nivel 3 o Nivel Definido:

La organización entera participa en el proceso eficiente de proyecto software.

- Se conoce de antemano los procesos de construcción de software
- Existen métodos y plantillas bien definidas y documentados
- Los procesos no solo afectan a los equipos de desarrollo sino a toda la organización relacionada
- Los proyectos se pueden definir cualitativamente

Nivel 4 o Nivel Cuantitativamente Administrado:

Se puede seguir con indicadores numéricos (estadísticos) la evolución de los proyectos.

- Las estadísticas son almacenadas para aprovechar su aportación en siguientes proyectos
- Los proyectos se pueden pedir cuantitativamente

Nivel 5 o Nivel Optimizado:

En base a criterios cuantitativos se pueden determinar las desviaciones más comunes y optimizar procesos.

- En los siguientes proyectos se produce una reducción de costes gracias a la anticipación de problemas y la continua revisión de procesos conflictivos

Paso de un nivel inferior a otro nivel superior:

Para pasar de Nivel 0 a Nivel 1 es necesario:

- Concluir el proceso de desarrollo

Para pasar de Nivel 1 a Nivel 2 es necesario:

- Agregar los procesos:
 - Gestión de requisitos
 - Planificación de proyectos
 - Seguimiento y control de proyectos
 - Gestión de proveedores
 - Aseguramiento de la calidad
 - Gestión de la configuración

Para pasar de Nivel 2 a Nivel 3 es necesario:

- Agregar los procesos:
 - Desarrollo de requisitos
 - Solución Técnica
 - Integración del producto
 - Verificación
 - Validación
 - Desarrollo y mejora de los procesos de la organización
 - Definición de los procesos de la organización
 - Planificación de la formación

- Gestión de riesgos
- Análisis y resolución de toma de decisiones

Para pasar de Nivel 3 a Nivel 4 es necesario:

- Agregar los procesos:
 - Gestión cuantitativa de proyectos
 - Mejora de los procesos de la organización

Para pasar de Nivel 4 a Nivel 5 es necesario:

- Agregar los procesos:
 - Innovación organizacional
 - Análisis y resolución de las causas

Beneficios de CMMI

Aplicando correctamente este modelo, se obtienen beneficios, entre los que destacan los siguientes:

Beneficios del ámbito de la ingeniería:

1. Los procesos maduros permiten:
 - Comprender lo que está pasando
 - Que el personal desarrolle todo su potencial de forma más completa y efectiva dentro de la organización
2. La mejora de los procesos tiene más posibilidades de resultar con éxito y ser más sustanciosa a la organización ya que se basa en la definición, medición, y control de los procesos.
3. Se incrementa sensiblemente la probabilidad de éxito en la introducción de nuevas y apropiadas tecnologías, técnicas y herramientas en la organización.

Beneficios de ámbito económico y organizativo:

1. Enfatiza el desarrollo de procesos en las organizaciones, que permiten mejorar el desarrollo de los productos y los servicios ofertados a los clientes.
2. Proporciona un marco de trabajo que permite organizar y priorizar las actividades de mejora de procesos, involucrando al propio producto, al negocio, al personal, y a la tecnología.
3. Da soporte a la coordinación de actividades multidisciplinarias que pueden ser necesarias para construir con éxito un determinado producto.
4. Enfatiza el alineamiento de los objetivos de la mejora de procesos con los objetivos de negocio de las organizaciones.

2.5 ¿Qué es MOF?

MOF (*Microsoft Operation Framework*). (Marco de Referencia de Operaciones de Microsoft). MOF es un modelo que surge a la sombra de una gran empresa tecnológica, para dar cobertura a la implantación de sus propios productos. Aunque esto no debe limitar su uso dado que sus preceptos pueden ser aislados.

MOF ofrece directrices sobre el modo de planificar, implementar y mantener procesos operativos de TI que respalden las soluciones de servicio críticas. MOF es un modelo genérico y, por este motivo, debe adaptar muchas de las recomendaciones para usarlas en su empresa. Cuando se encuentran referencias a "funciones" en el modelo MOF, hay que tener en cuenta que se puede asignar a una misma persona a varias funciones, sobre todo en las empresas pequeñas. No obstante, aunque represente a todo el departamento de TI, los procedimientos y recomendaciones de este modelo se pueden aplicar de forma general.



Ilustración 6. Modelo MOF

MOF es un modelo estructurado y flexible que se basa en:

- Los equipos de consultoría y soporte técnico de Microsoft y su experiencia de trabajo con clientes empresariales y socios, además de grupos internos de operaciones de TI en Microsoft.
- La Biblioteca de infraestructuras de TI (ITIL), que describe los procesos y las prácticas recomendadas necesarios para el suministro de soluciones de servicio críticas.
- ISO/IEC 15504, de la Organización Internacional de Normalización (ISO), que proporciona un enfoque normalizado para evaluar la madurez del proceso de software.

MOF ofrece recomendaciones para la implementación de varios productos de Microsoft, como Microsoft Windows Server 2003 y Microsoft Exchange Server 2007.

MOF complementa y se integra en Microsoft Solutions Framework (MSF). MSF es un enfoque por disciplinas para la administración de proyectos tecnológicos basados en prácticas internas de Microsoft, la experiencia del Servicio de soporte técnico de Microsoft con clientes y socios, y las prácticas recomendadas del sector para el desarrollo de software y la administración de proyectos. MSF es un enfoque para diseño e implementación de sistemas de TI (por ejemplo, un proyecto de migración para pasar de Lotus Notes a Exchange Server 2007), mientras que MOF trata la administración diaria de un sistema o un entorno como, por ejemplo, una organización de Exchange Server 2007.

El modelo de proceso MOF está formado por cuadrantes, revisiones de la administración de las operaciones y revisiones de la administración de los servicios.

2.6 ¿Qué es COSO?

COSO (*Committee of Sponsoring Organizations of the Treadway Commission*). Originariamente era conocido como el informe COSO, aunque en la actualidad es referido como COSO-ERM o COSO II. Se trata de un modelo de referencia para buenas prácticas que define el control interno como un proceso, es decir, como algo que ocurre y evoluciona en el tiempo cuya responsabilidad recae en todas las personas de una organización y que proporciona una seguridad razonable para el logro de los objetivos de una empresa.

Los objetivos se catalogan en cuatro categorías:

1. Estratégica (Strategic): relativo a los objetivos más generales de la organización que están alineados con la misión de la misma.
2. Operativo (Operational): se basa en obtener la eficacia y eficiencia de los recursos y activos.
3. Reporte (Reporting): comprende la fiabilidad de la información, normalmente la financiera.
4. Cumplimiento (Compliance): trata sobre el cumplimiento de leyes y normas que sean aplicables.



Ilustración 7. Cubo COSO con sus áreas conectadas

Está compuesto básicamente por:

- Entorno de control
- Evaluación de los riesgos
- Actividades de control
- Información y comunicación
- Supervisión

La principal diferencia entre COSO y COBIT es que COSO está enfocado a toda la organización mientras que COBIT está centrado en el entorno TI.

2.7 Diferencias de los distintos modelos

Como hemos visto hasta ahora existen multitud de metodologías dentro del ámbito del gobierno de una organización, más aún de las que aquí se han citado, algunas de ellas no tanto del gobierno TI como tal, sino del gobierno en general. Pero las expuestas son algunas de las más usadas y conocidas internacionalmente.

Muchas de ellas tienen puntos en común, aspectos diferentes especializados por algún tipo de área, y en la mayoría de casos compatibles entre ellas. Todas las metodologías están orientadas a la realización de unas buenas prácticas en el gobierno empresarial.

Por ejemplo, ITIL está orientado a la administración y gestión de servicios TI. En cambio CMMI está orientado al desarrollo del software. COSO por su parte está enfocado a toda la organización más que a las TI en particular, al contrario que COBIT, que es la que más ámbitos abarca en la gestión de las TI aparte de ser un referente para temas de auditoría y medición de las TI.

La elección de uno o varios modelos depende del tipo de organización con la que se esté tratando. En algunas se podría utilizar unas que no en otras o varias que sean compatibles. Se debe tener en cuenta también que la implantación de estos modelos puede resultar complejo y no ver resultados positivos a corto plazo pero que en el futuro seguramente se conviertan casi en imprescindibles y con valor para la organización.

En cualquier caso siempre se debe pensar en que estos son modelos que tratan de llevar un control de la empresa para facilitar los procesos y en definitiva para ayudar a conseguir los objetivos y puede que cumplir con unas leyes, por lo que la aplicación de estos modelos debe seguir una lógica, deben ser implantados con lógica y realismo, y no convertirse en lugar de ayudar, en un problema para la empresa.

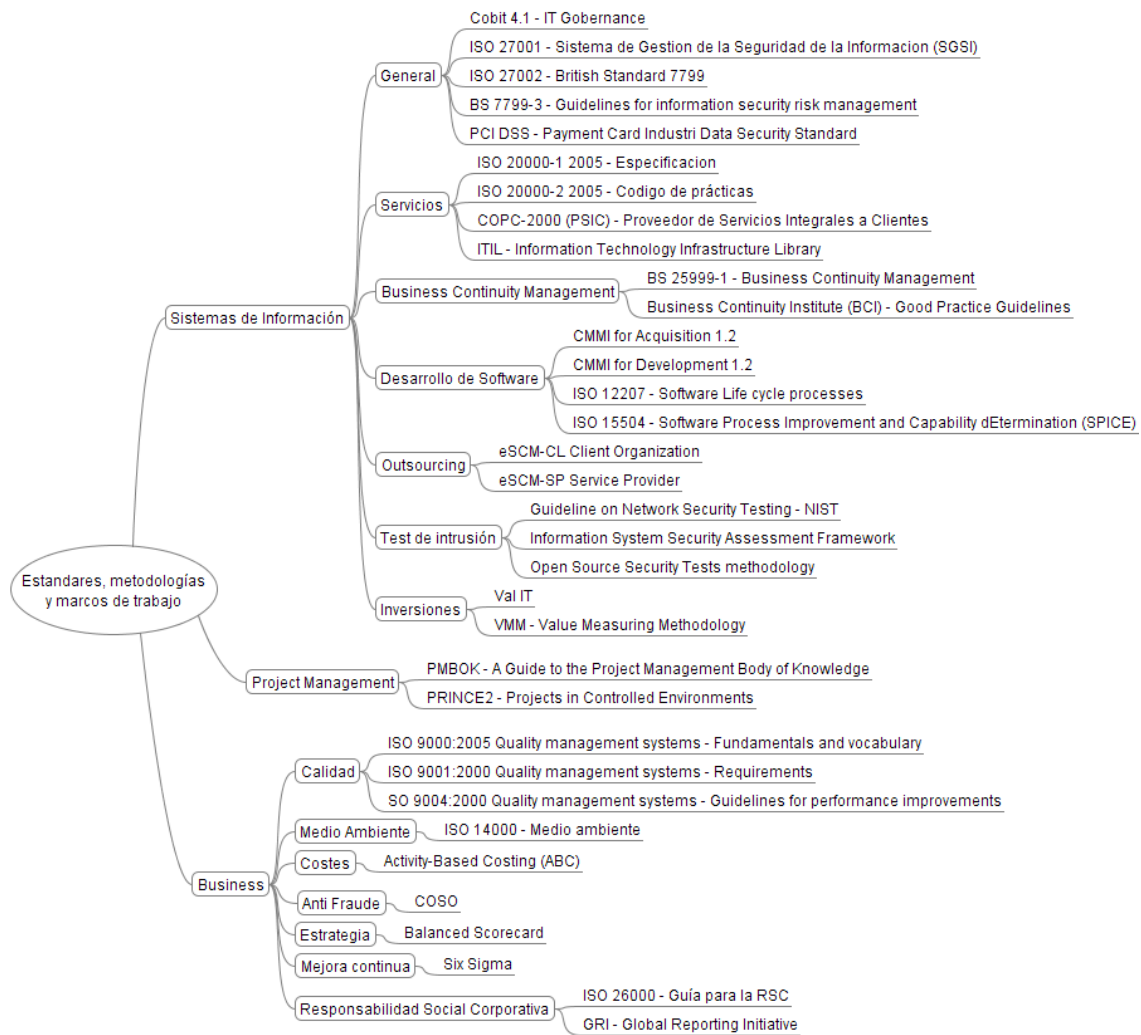


Ilustración 8. Mapa de los distintos estándares, metodologías y marcos de trabajo existentes

2.8 Certificaciones ISO

Los modelos sobre gestión de gobierno TI vistos hasta ahora (COBIT, ITIL, CMMI, MOF) son estándares desde el punto de vista de que sirven como tipo, patrón o referencia para la realización de buenas prácticas TI en una empresa. Pero estos pueden ser implantados de muchas maneras distintas en las organizaciones y no son medibles.

Las normas ISO (International Organization for Standardization) son un conjunto de normas, reglas y pautas que toda empresa debe cumplir si quiere lograr una certificación o certificado ISO que corrobore que se cumplen los preceptos establecidos para otorgar, en el caso TI, servicios de calidad demostrados y comprobados. Tienen el respaldo de los organismos de normalización internacionales, lo que ayuda a que se asienten como un referente. Además, se han publicado también como norma nacional en muchos países y cumplen todo lo necesario para que los gobiernos puedan incluirlas en sus legislaciones (pudiendo llegar a convertirse en obligatorias).

Es decir, si implantamos un modelo como ITIL o cualquier otro, este no está realmente garantizando calidad en una empresa aunque sea posible que la tenga, en cambio con una certificación ISO, las organizaciones son auditadas y pueden ser medidas frente a un conjunto establecido de requisitos y por tanto una certificación ISO supone un elemento diferenciador en una empresa ya que es una distinción demostrada de calidad. También hay que decir que el lograr una certificación ISO seguramente pase por estar realizando correctamente unas buenas prácticas con alguno o varios modelos de gobierno TI.

Las principales ISO relacionadas con el ámbito TI son:

ISO 9001 (calidad), ISO 20000 (servicio), ISO 27001 (seguridad).

ISO 9001. Gestión de Calidad.

Es un conjunto de normas sobre la calidad y la gestión. La Norma ISO 9001 ha sido elaborada por el Comité Técnico ISO/TC176 de ISO Organización Internacional para la Estandarización y especifica los requisitos para un buen sistema de gestión de la calidad que pueden utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales.

La norma ISO 9001 tiene origen en la norma BS 5750, publicada en 1979 por la entidad de normalización británica, la British Standards Institution (BSI).

La versión actual de ISO 9001 (la cuarta) data de noviembre de 2008, y por ello se expresa como ISO 9001:2008. Versiones ISO 9001 hasta la fecha:

Cuarta versión: la actual ISO 9001:2008 (15/11/2008)

Tercera versión: ISO 9001:2000 (15/12/2000)

Segunda versión: ISO 9001:94 - ISO 9002:94 - ISO 9003:94 (01/07/1994)

Primera versión: ISO 9001:87 - ISO 9002:87 - ISO 9003:87 (15/03/1987)

En la primera y segunda versión de ISO 9001, la Norma se descomponía en 3 normas: ISO 9001, ISO 9002, e ISO 9003.

ISO 9001: organizaciones con diseño de producto

ISO 9002: organizaciones sin diseño de producto pero con producción/fabricación.

ISO 9003: organizaciones sin diseño de producto ni producción/fabricación (comerciales).

El contenido de las 3 normas era el mismo, con la excepción de que en cada caso se excluían los requisitos de aquello que no aplicaba. Esta mecánica se modificó en la tercera versión, unificando los 3 documentos en un único estándar, sobre el cual se realizan posteriormente las exclusiones. La cuarta versión de la norma presenta más de 60 modificaciones.

ISO 20000. Gestión del Servicio. Basada en ITIL.

La serie ISO/IEC 20000 - Service Management normalizada y publicada por las organizaciones ISO e IEC (International Electrotechnical Commission) el 14 de diciembre de 2005, es el estándar reconocido internacionalmente en gestión de servicios de TI. La serie 20000 proviene de la adopción de la serie BS 15000 desarrollada por la entidad de normalización británica, la British Standards Institution (BSI).

La gestión de una entrega efectiva de los servicios de TI es crucial para las empresas. Hay una percepción de que estos servicios no están alineados con las necesidades y requisitos del negocio. Esto es especialmente importante tanto si se proporciona servicios internamente a clientes como si se está subcontratado proveedores. Una manera de demostrar que los servicios de TI están cumpliendo con las necesidades del negocio es implantar un Sistema de Gestión de Servicios de TI (SGSTI) basado en los requisitos de la norma ISO/IEC 20000. La certificación en esta norma internacional permite demostrar de manera independiente que los servicios ofrecidos cumplen con las mejores prácticas.

ISO/IEC 20000 está basada y reemplaza a la BS 15000, la norma reconocida internacionalmente como una British Standard (BS), y que está disponible en dos partes: una especificación auditable y un código de buenas prácticas.

La ISO/IEC 20000 es totalmente compatible con la ITIL (IT Infrastructure Library), o guía de mejores prácticas para el proceso de GSTI. La diferencia es que el ITIL no es medible y puede ser implantado de muchas maneras, mientras que en la

ISO/IEC 20000, las organizaciones deben ser auditadas y medidas frente a un conjunto establecido de requisitos.

La ISO/IEC 20000 es aplicable a cualquier organización, pequeña o grande, en cualquier sector o parte del mundo donde confían en los servicios de TI. La norma es particularmente aplicable para proveedores de servicios internos de TI, tales como departamentos de Información Tecnológica, proveedores externos de TI o incluso organizaciones subcontratadas. La norma está impactando positivamente en algunos de los sectores que necesitan TI tales como subcontratación de negocios, Telecomunicaciones, Finanzas y el Sector Público.

La ISO/IEC 20000 está dividida en las siguientes secciones que definen los requisitos que debe cumplir una organización, la cual proporciona servicios a sus clientes con un nivel aceptable de calidad:

- Requisitos para la gestión de un sistema.
- Implantación y planificación de Gestión de Servicios.
- Planificación e implantación de servicios nuevos o modificados.
- Procesos del servicio de entrega.
- Procesos relacionales.
- Procesos de control.
- Procesos de emisión.

Demuestra que se tienen procedimientos y controles adecuados in situ para proporcionar un servicio de calidad de TI coherente y a un coste efectivo.

Los suministradores de servicios de TI se han vuelto cada vez más sensibles y responsables con los servicios que prestan más que de la tecnología que puedan proporcionar.

Los proveedores externos de servicios pueden usar la certificación como un elemento diferenciador y acceder a nuevos clientes, ya que esto cada vez más se convierte en una exigencia contractual.

Permite seleccionar, gestionar y proporcionar un servicio externo más efectivo. Ofrece oportunidades para mejorar la eficiencia, fiabilidad y consistencia de sus servicios de TI que impactan positivamente tanto en los costes como en el servicio.

La aparición de la serie ISO/IEC 20000, ha supuesto el primer sistema de gestión en servicio de TI certificable bajo norma reconocida a nivel mundial. Hasta ahora, las organizaciones podían optar por aplicar el conjunto de mejoras prácticas dictadas por ITIL (completadas por otros estándares como CMMI o COBIT) o certificar su gestión contra el estándar local británico BS 15000.

ISO 27001. Gestión de Seguridad.

El estándar para la seguridad de la información ISO/IEC 27001 fue aprobado y publicado como estándar internacional en octubre de 2005 por la International

Organization for Standardization y por la comisión International Electrothechnical Commission.

Especifica los requisitos necesarios para establecer, implantar, mantener y mejorar un Sistema de Gestión de la Seguridad de la Información (SGSI) según el conocido “Ciclo de Deming”: PDCA, acrónimo de Plan, Do, Check, Act (Planificar, Hacer, Verificar, Actuar).

Es consistente con las mejores prácticas descritas en ISO/IEC 17799 (actual ISO/IEC 27002) y tiene su origen en la norma BS 7799-2:2002, desarrollada por la entidad de normalización británica, la BSI.

El equipo de proyecto de implantación de una ISO 27001 debe estar formado por representantes de todas las áreas de la organización que se vean afectadas por el SGSI, liderado por la dirección y asesorado por consultores externos especializados en seguridad informática generalmente Ingenieros o Ingenieros Técnicos en Informática, derecho de las nuevas tecnologías, protección de datos y sistemas de gestión de seguridad de la información (que hayan realizado un curso de implantador de SGSI).

2.9 Otros Conceptos

Misión, visión, objetivos, metas, métricas y acciones

Las empresas deben basarse en un plan de sistemas y TI para lograr su visión/misión, esto es, para llegar a una situación a la que la empresa mira como orientación. La visión podría compararse a un sueño, algo a lo que deseamos llegar, y la misión podría compararse con el propósito que queremos lograr.

Mediante un análisis interno y externo o análisis de entorno, las empresas pueden conocer sus fortalezas y debilidades en el mercado y sacar conclusiones y también conocer a qué factores deben hacer frente como a las restricciones originadas por la política y la estructura propia de la empresa y restricciones del entorno de la empresa como el país en el que se encuentren con sus leyes, etc.

Para llegar a conseguir la visión debemos alcanzar la misión y para ello la empresa necesita definir una estrategia, unos objetivos estratégicos que nos den la dirección de lo que tenemos que conseguir, con la ayuda del análisis interno y externo.

Una vez definidos los objetivos, para saber lo que hay que hacer y cómo y que los cumplimos correctamente, la empresa debe hacer uso de métricas, metas y acciones. Las acciones son las tareas como tales, lo que hay que hacer. Las metas son los resultados que esperamos obtener. Y las métricas nos van a decir el resultado obtenido, el control del nivel de logro. Por ejemplo al realizar una serie de acciones alcanzaremos unas metas y sabremos si esas metas han sido positivas o negativas mediante unas métricas.



Ilustración 9. Planificación estratégica de la empresa

Por ejemplo imaginemos el escenario de una persona que finaliza sus estudios de Bachillerato y quiere seguir estudiando y le encanta todo lo relacionado con los ordenadores y las tecnologías. Imaginemos que esta persona es una empresa. La visión o sueño de esta persona podría ser el de 'ser el mayor experto en tecnología'. Su misión es su propósito, es decir lograr su visión, por tanto su misión es la de 'convertirse en un experto en tecnología'. Mediante un análisis interno podría conocer si tiene suficientes conocimientos en matemáticas, si es mejor en lógica o en literatura. Y mediante un análisis externo sabría si puede estudiar donde él vive por ejemplo. Una vez que sabemos lo que queremos ser, pensaríamos en los objetivos para lograrlo, es decir, definir adonde queremos llegar y los plazos para conseguirlo. Por tanto el

objetivo de esta persona sería el de estudiar una carrera de ingeniería, por ejemplo ingeniería informática, en 4 años. Y para cumplir este objetivo realizaría una serie de acciones, metas y métricas. Las acciones serían matricularse en la carrera, asistir a clases... Las metas serían los resultados esperados por ejemplo ir sacando buenas notas, serían buenas metas para esta persona, ya que son el fin deseado que le van acercando cada vez más a su objetivo. Y las métricas podrían ser las calificaciones obtenidas en los exámenes, serían un buen indicador de cómo esta persona está evolucionando hacia su objetivo.

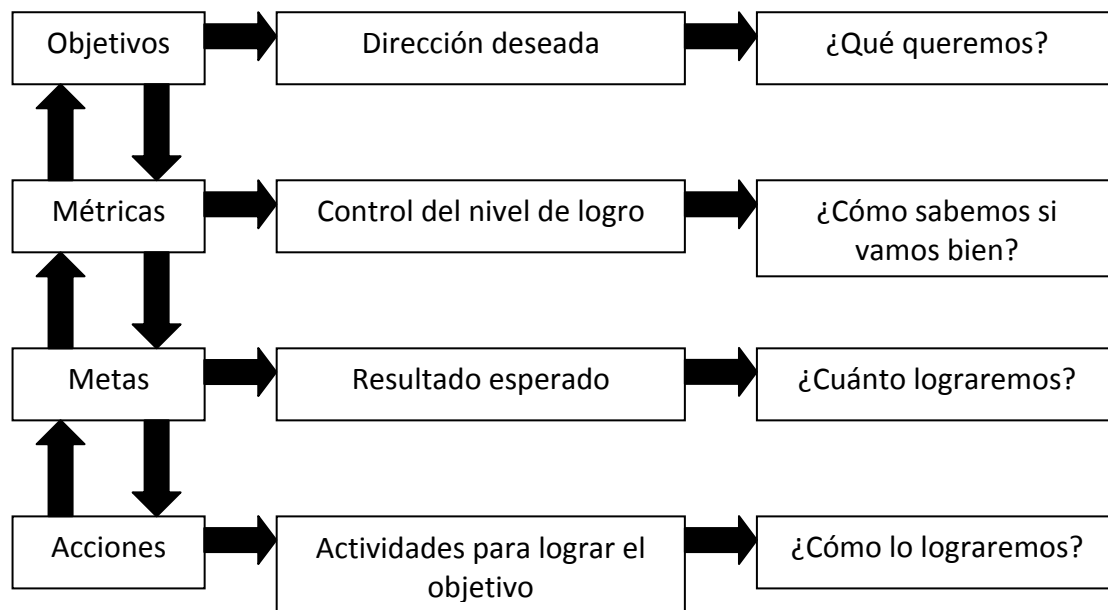


Ilustración 10. Objetivos, métricas, metas y acciones

Objetivos de negocio, Objetivos TI, Objetivos de control

Dentro de los objetivos estratégicos nos encontramos con objetivos de distintos tipos como son los objetivos de negocio, los objetivos TI y los objetivos de control.

Los objetivos de negocio son los objetivos que una empresa, por medio de su dirección, tienen como fin. Es decir es el objetivo más genérico, de más alto nivel, que una empresa tiene en su mente para prosperar.

Por otro lado, los objetivos TI son los objetivos que una empresa desea realizar para obtener sus objetivos de negocio ayudándose de las tecnologías de la información. Esto es, las TI deben ayudar a conseguir los objetivos de negocio, es por ello que es tan importante que ambos objetivos se encuentren en perfecta alineación.

Los objetivos de control son los objetivos de más bajo nivel, es decir de nivel de detalle. Esto es, los objetivos detallados para conseguir tener unos buenos objetivos TI con su control interno y en consecuencia alcanzar el objetivo de negocio empresarial.

Por ejemplo, una empresa puede tener como objetivo de negocio: incrementar el número de clientes. Este es un objetivo lógico de una organización para aumentar su capacidad comercial e incrementar su valor económico.

¿Cómo incrementar clientes? Haciendo uso de las TI una empresa podría implantar un CRM (Customer Relationship Management, Gestión de las relaciones con los clientes). El CRM, también conocido como marketing relacional, es un modelo de gestión de las empresas encargado de la orientación de la organización hacia el cliente. Un buen uso de un modelo CRM implantado puede hacer mejorar la relación del cliente con la empresa por ejemplo mediante un eficaz *call center*, incluido dentro de un CRM, un centro de llamadas que atienda las peticiones de los clientes.

Para implantar un CRM tendríamos que cumplir con los objetivos de control. Estos objetivos son los encargados de velar por el cumplimiento de las actividades tecnológicas por medio de acciones, métricas, etc.

El correcto funcionamiento de estos objetivos hará que la empresa se encuentre bien alineada y como resultado será más probable que la organización fructifique en el logro de su objetivo de negocio a través del cual conseguirá su misión, se conseguirá tomar la dirección deseada de lo que se quiere.

Modelos de conformidad

Gobierno, Gestión de riesgos y Conformidad (Governance, Risk Management, and Compliance GRC) es el abanico de términos involucrados dentro de una organización. Cada día estas tres áreas están empezando a estar más integradas y alineadas con el fin de evitar conflictos e inútiles solapamientos. Mientras que GRC es interpretada de forma distinta en las organizaciones, las actividades típicas que se aplican son el gobierno corporativo, la gestión del riesgo empresarial y el cumplimiento de las leyes y reglamentos.

Gobierno: describe el enfoque global de la gestión a través de la alta dirección de una empresa que dirigen y controlan toda la organización, combinando la gestión de la información y las estructuras jerárquicas de control de la gestión. Actividades típicas de un gobierno son el asegurar que la información sea completa, precisa y oportuna para permitir la toma adecuada de decisiones y proporcionar los mecanismos de control necesarios para asegurar que las estrategias, directrices e instrucciones de la dirección de una empresa se llevan a cabo de manera sistemática y efectiva.

Gestión del riesgo: es el conjunto de procesos mediante los cuales la administración identifica, analiza y cuando es necesario responde adecuadamente a los riesgos que pudieran afectar a la realización de los objetivos de negocio de la organización. La respuesta a los riesgos por lo general depende de la gravedad percibida, entendiéndose como riesgos en una empresa los riesgos tecnológicos, riesgos comerciales o financieros, información de riesgos de seguridad, riesgos externos de cualquier índole, en definitiva una amplia gama de riesgos que las

organizaciones deben controlar y evitar, y con mayor razón si cabe desde la aparición de leyes como la ley SOX en EEUU.

Conformidad: El cumplimiento de los medios conformes a los requisitos establecidos a nivel organizativo se logra a través de los procesos de gestión que permiten identificar los requisitos aplicables, por ejemplo definiendo en las leyes, reglamentos, contratos, estrategias y políticas. Es por tanto importante evaluar el estado del cumplimiento, evaluar los riesgos y los costos, y priorizar, financiar y poner en marcha las medidas correctoras que se consideren necesarias. Todo esto debería estar reflejado en leyes, normas, y estrategias empresariales.

La finalidad es por tanto, aparte de un control para evitar escándalos financieros de las organizaciones por parte de leyes como la SOX, añadir valor al negocio a través de la toma de decisiones que mejoren la estrategia y operatividad de la planificación en la empresa. Por ello todo lo que tiene que ver con el gobierno TI va más allá de un control de auditoría de empresas, una buena gestión de gobierno TI incrementará el valor de las organizaciones ya que será más fácil mejorar en la eficacia y eficiencia de las actividades que se realizan siguiendo un correcto enfoque.

Acuerdo de Nivel de Servicio (ANS), Service Level Agreement (SLA)

Los ANS son un aspecto muy importante dentro del ámbito de la empresa con sus clientes y proveedores dentro del campo del gobierno TI ya que es en ellos donde se van a explicar detalladamente todas las características que un cliente desea como servicio. Muchos problemas derivados de los productos o servicios TI que se realizan a los clientes vienen precisamente por no establecer claramente los requisitos o por cambios en la estructura con proyectos ya avanzados. Mediante los ANS se intenta por tanto evitar este tipo de problemas a la hora de realizar proyectos TI.

Un servicio es algo diferente a la simple compraventa de bienes, es una prestación de una empresa a otra. Por ejemplo, en la venta de un ordenador, explicar las características de dicho ordenador que se va a vender es un servicio que estamos dando.

- ¿Qué ocurre si no se regulan los servicios?
- Que nos preguntaríamos varias cosas como:

¿Está el servicio incluido o no?, ¿el volumen crece o disminuye?, ¿hay un sistema para tratar incidencias?, ¿es bueno o malo el servicio?, ¿cuándo se está cumpliendo?, ¿cuál es el coste real?

La empresa puede que tenga la necesidad de subcontratar a otras empresas (*outsourcing*) porque desconoce lo que le está costando dar estos servicios con la pérdida económica que ello supone. Pero además, todo esto provoca quejas de los clientes porque no se mide el nivel de satisfacción del usuario final. Sin controlar el servicio el cliente podría sentir que:

- No le atienden todo lo que pide.

- Los resultados no son los que necesita.
- Tardan más de lo necesario.
- Cuando surge un problema parece no haber nadie que le ayude.

Y no solo eso, sino que en un mundo sin regulación de servicios, también pueden surgir las quejas en los proveedores. Ya que estos pueden pensar que están creando servicios que no son de su responsabilidad.

Una manera de controlar el servicio que la empresa está creando es realizar un Acuerdo de Nivel de Servicio (ANS). Un ANS es un conjunto de indicadores anexo al contrato en el cual se pactan los niveles de prestación y calidad entre proveedor y cliente, así controlamos los elementos indicadores de volumen. Está formado por:

1. Alcance
2. Condicionamiento
3. Indicadores
4. Reglas
5. Precio (de ofrecer el servicio al proveedor)
6. Penalizaciones (cláusulas que dan credibilidad) (por ejemplo: resolver problemas en 24 horas o en 3 días...)

El ANS consta en resumen de: definición del servicio, acuerdo explícito, medición y responsabilidades.

El ANS se separa muy poco del contrato, porque está todo muy definido, faltan las cláusulas que convierten el documento en uno legal, es decir, quiénes intervienen y qué ocurre en caso de incumplimiento legal.

El ANS se aplica en:

- Servicios TI
- Servicios outsourcing
- Servicios help desk
- Servicios de edición de documentos
- Servicios de mensajería y distribución

Cuadro de Mando Integral CMI, Balanced Scorecard (BSC)

El cuadro de mando es una herramienta de gestión que facilita la toma de decisiones y recoge un conjunto de indicadores que proporcionan a la alta dirección una visión del estado de una organización. La información obtenida permite tomar decisiones para aumentar la calidad en los procesos de una empresa. Un cuadro de mando dependiendo al área al que nos queramos dedicar tendrá unos indicadores distintos que otros cuadros de mando. Sin embargo el cuadro de mandos integral es el que recoge todo los niveles de un negocio.

Modelo de Madurez

Un modelo de madurez consiste en un proceso mediante el cual vamos a saber como es el camino de una organización para alcanzar buenas prácticas en la gestión de proyectos a través de distintos niveles de madurez. El modelo de madurez explicará en que punto nos encontramos en nuestra capacidad de gerencia existente en la organización.

La mayoría de modelos de madurez están basados en el modelo de madurez de desarrollo de software Capability Maturity Model (CMM). El CMM es un modelo de evaluación de los procesos en una organización. Fue inicialmente desarrollado para funciones relativas al desarrollo e implementación de software por la Universidad Carnegie-Mellon para el Software Engineering Institute (SEI).

Este modelo establece un conjunto de buenas prácticas divididas por Áreas Clave de Proceso (KPA, Key Process Area). Las áreas se refieren a las prácticas de la organización como:

- Definidas en un procedimiento documentado
- Provistas (la organización) de los medios y formación necesarios
- Ejecutadas de un modo sistemático, universal y uniforme (institucionalizadas)
- Medidas
- Verificadas

Estas áreas se agrupan en cinco niveles de madurez que serán los que nos digan como de madura está la organización en cada área. Los niveles son

Los niveles son:

1. **Inicial:** Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Existe una falta de planificación y aunque se usen técnicas correctas de ingeniería los esfuerzos se ven en la mayoría de los casos ineficientes produciéndose fracasos, retrasos y sobrecostos. El resultado de los proyectos es impredecible.
2. **Repetible:** En este nivel existen unas métricas básicas y un razonable seguimiento de la calidad. Se dispone además de unas prácticas institucionalizadas de la gestión de proyectos así como una gestión sistemática de la relación con subcontratistas y clientes.
3. **Definido:** Existen correctos procedimientos de coordinación entre grupos, técnicas de ingeniería más detalladas, formación de personal y un nivel más avanzado de métricas en los procesos.
4. **Gestionado:** Se caracteriza porque las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.

5. **Optimizado:** La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Los modelos de madurez por lo tanto buscan establecer una valoración estandarizada que nos permita conocer el estado de una empresa en sus distintas áreas o procesos.

CMMI es una integración de varios CMM existentes y COBIT tiene su propio modelo de madurez basado en 6 niveles de madurez.

Assessments (Evaluaciones)

Los llamados 'Assessments' son evaluaciones o cuestionarios, es decir una batería de tests, que a modo de una métrica nos ayudan a conocer el nivel de madurez en este caso el nivel de madurez que tiene una empresa en el uso de las TI.

3. Arquitectura Tecnológica

3.1 Modelo de arquitectura

La arquitectura de software elegida para el presente proyecto ha sido el Modelo Vista Controlador (MVC) ya que es el que mejor se adapta a las características de la aplicación realizada. MVC es un patrón de arquitectura que separa los datos de la aplicación, la interfaz de usuario y la lógica en tres partes distintas.

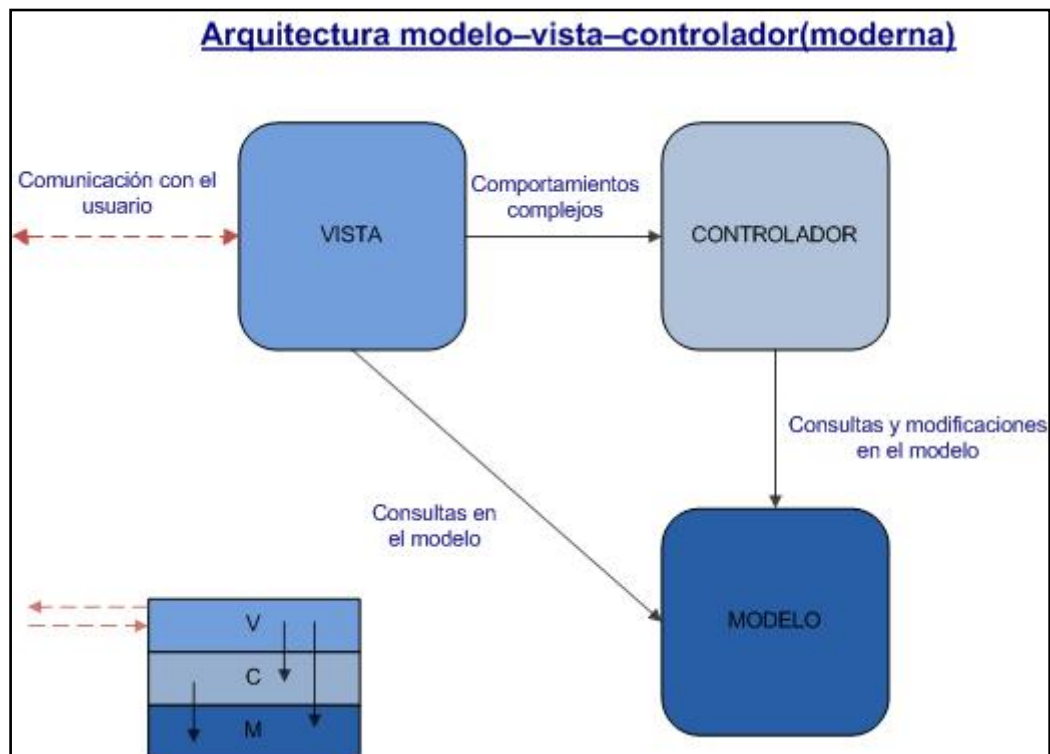


Ilustración 11. Modelo vista-controlador

Los componentes de la arquitectura MVC son:

Modelo: esta capa recoge el modelo conceptual del sistema. En él se incluirán los subsistemas y clases relacionadas con los datos que maneja el sistema. El componente Modelo no contiene ninguna referencia hacia los otros dos componentes Vista y Controlador. El propio sistema es el encargado de enlazar los componentes Modelo y Vista, es decir, cualquier cambio que se produzca en el Modelo se ha de notificar a la Vista.

Vista: es la capa con la que interactúa el usuario directamente. El usuario visualiza los datos recogidos en el modelo conceptual a través de la Vista. La Vista está compuesta de las interfaces gráficas que el usuario podrá utilizar para manejar los datos del

sistema. La Vista se comunica con el Controlador para realizar una acción mandada por el usuario.

Controlador: capa que responde a eventos producidos por la interacción del usuario con el sistema. Es la encargada de gestionar dichos eventos (recibidos de la Vista) de tal forma que traduce los cambios que provocan estos eventos al modelo conceptual (capa Modelo). Según lo expuesto anteriormente cabe decir que la capa Controlador está relacionada tanto con la Vista como con el Modelo.

En la aplicación del proyecto, la vista es la interfaz con el usuario, es decir las ventanas propias de la aplicación que el usuario utiliza. El modelo es el sistema de gestión de la base de datos y la lógica en Java. Y el controlador es el responsable de recibir los eventos de entrada del usuario desde la vista y devolver eventos o datos basándose en el modelo lógico de la aplicación.

3.2 Software utilizado

Todas las aplicaciones y librerías usadas en el proyecto son de libre uso y gratuitas por tener licencia pública o estar permitido su uso para fines académicos, a excepción de Microsoft Word (dentro del paquete de Microsoft Office Professional 2003-2010) que es el único que no está disponible gratuitamente pero del que se disponen copias en las aulas informáticas de la Universidad Carlos III para su utilización y a excepción también de Altova MissionKit que ofrece una aplicación de prueba de 30 días de uso. Los sistemas operativos de Microsoft así como las aplicaciones Microsoft Visio Professional 2010 y Microsoft Project Professional 2003-2010 utilizados son gratuitos debido a que Microsoft mantiene un acuerdo a través del MSDN Academic Alliance (conocido ahora como DreamSpark) con las universidades y provee a los estudiantes de su uso gratuito con fines académicos. Linux Ubuntu es libre y gratuito con licencia GPL (General Public License).

Para la realización del proyecto se ha utilizado el siguiente software:

Lenguajes de programación/Bases de Datos:

- **Java:** es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. Gran parte de la sintaxis de Java está basada en el lenguaje C y C++ pero con un modelo más simple eliminando algunas herramientas de bajo nivel que el lenguaje C permite y que pueden inducir a errores como la manipulación directa de punteros. Es uno de los más usados lenguajes de programación en la actualidad y su principal característica es que está basado en la POO (Programación Orientada a Objetos) lo cual significa que el diseño de las aplicaciones se hace mediante objetos y sus interacciones para conseguir códigos más legibles y lógicos. Algunas de las típicas técnicas de POO son la herencia, abstracción, polimorfismo y encapsulamiento. Java Swing es el término empleado para referirse a las aplicaciones con ventanas. Swing es el nombre dado a la biblioteca de Java para tratar la interfaz gráfica. Java es el lenguaje en el que está implementada la aplicación del proyecto.

Librerías de Java:

iText (iText-5.0.6.jar): es una librería con licencia gratuita para la manipulación y creación de archivos PDF. Se ha utilizado en la creación de la solución para el guardado de ficheros PDF.

Ojdbc6 (ojdbc6.jar): es una librería gratuita (JDBC) que Oracle pone a disposición para permitir la conexión entre Java y una base de datos en este caso una base de datos de Oracle. Java DataBase Connectivity (JDBC) es una API (conjunto de métodos de una biblioteca o librería) que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java,

independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede.

- **SQL (Structured Query Language):** es un lenguaje de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar información de interés de una base de datos, así como también hacer cambios sobre ella. La base de datos usada en el proyecto está escrita en lenguaje SQL.

Aplicaciones Base de Datos:

- **Oracle 10g:** Oracle es un sistema de gestión de base de datos objeto-relacional (conocido también como ORDBMS, Object-Relational Data Base Management System), desarrollado por Oracle Corporation. En el proyecto se ha utilizado la versión Oracle 10g. Entre las distintas aplicaciones que ofrece el paquete Oracle 10g se ha utilizado SQL Plus y el Database Control.
SQL Plus: es un programa de línea de comandos de Oracle que puede ejecutar comandos SQL y PL/SQL de forma interactiva o mediante un script.
Database Control: se usa para administrar de forma general la base de datos desde un entorno web.

Aplicaciones edición programación:

- **Notepad ++:** es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Es similar al Bloc de notas de Windows en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores. En nuestro caso se ha usado para la realización de los scripts que ejecutan la base de datos. También como editor de algunas sintaxis de Java.
- **NetBeans IDE 6.9.1:** Es un entorno de desarrollo integrado (IDE) disponible para sistemas Windows, Mac, Linux y Solaris. Se trata de un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear programas usando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby on Rails, Groovy y Grails y C/C++. Se ha usado para el desarrollo de la aplicación creada en Java.
- **jGRASP:** Es un entorno de desarrollo ligero, creado específicamente para proporcionar la mejora de la visualización y comprensión del software. Está implementado en Java, y funciona en todas las plataformas con una Máquina Virtual Java. Se trata pues de un depurador y mesa de trabajo para la programación en lenguaje Java. Se ha utilizado para algunas comprobaciones de ejecución de código Java de la aplicación creada.

Aplicaciones edición de esquemas de diseño de Base de Datos/Ingeniería de Software:

- ***Microsoft Visio Professional 2010:*** es un software de dibujo vectorial para Microsoft Windows que permite dibujar diagramas de bases de datos, diagramas UML, diagramas de flujo de programas, diagrama de casos de usos... que permiten a los desarrolladores de lenguajes de programación diseñar esquemas visuales de sus aplicaciones. Se ha utilizado en el proyecto para la realización del diagrama EDT (Estructura de Desglose de Trabajo, en inglés WBS Work Breakdown Structure) y el diagrama de Casos de Uso.
- ***Microsoft Project Professional 2003-2010:*** es un software de administración de proyectos para ayudar a los administradores de proyectos a desarrollar planes de tiempo, de acciones, asignación de recursos a tareas, administración de presupuestos, análisis de cargas de trabajo y obtención de un seguimiento al progreso del proyecto. El diagrama Gantt del proyecto se ha realizado con esta herramienta de Microsoft.
- ***MySQL Workbench 5.2.38:*** es una herramienta para la administración y el diseño de bases de datos. Se ha utilizado para la creación del diagrama de Entidad Relación (E/R).
- ***Altova MissionKit:*** es una herramienta para el diseño de diagramas de bases de datos, XML y UML. Se ha usado para el diseño del diagrama UML (Unified Modeling Language/Lenguaje Unificado de Modelado).

Aplicaciones edición de texto:

- ***Microsoft Office Professional 2003-2010:*** es un paquete de aplicaciones de Microsoft que incluye entre otras a Word, Access, Excel, PowerPoint, FrontPage y Outlook. En el presente proyecto se ha utilizado Microsoft Word 2003-2010 que es una aplicación de creación y edición de archivos de texto.
- ***Adobe Acrobat Reader:*** es una herramienta que permite la visualización de archivos de texto en formato PDF. Los archivos en formato PDF son multiplataforma ya que pueden usarse en distintos sistemas operativos como Windows, Unix/Linux o Mac y además existe software libre para su manipulación.

Aplicación de estimación de costes de Software:

- ***Cocomo II 2000:*** el Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés CONstructive COSt MOdel) es un modelo matemático de base empírica utilizado para estimación de costes de software.

Aplicaciones edición de imagen:

- ***GIMP 2:*** es una aplicación libre y gratuita de edición de imágenes bajo licencia pública general de GNU.
- ***Microsoft Paint:*** es un programa básico de edición de imagen perteneciente a Microsoft.

Ambos editores se han usado para la realización y modificación de algunas imágenes del proyecto.

Internet.

Sistemas operativos:

- ***Microsoft Windows XP.***
- ***Microsoft Windows Vista.***
- ***Microsoft Windows 7.***
- ***Linux Ubuntu 10.***

3.3 Datos usados

La información utilizada desde la base de datos para el funcionamiento de la aplicación consta de tres partes diferenciadas. La primera sería la información obtenida desde COBIT 4.1, la segunda una selección de evaluaciones del IT Governance Network y la tercera un conjunto de problemas TI de fuente propia e Internet.

Los datos usados en la aplicación en la parte de la solución, así como de las consultas por la rama de la consulta por dominio y proceso COBIT, y en las partes de la administración de COBIT provienen del documento gratuito COBIT 4.1, que está disponible en www.isaca.org y cuyo copyright pertenece al IT Governance Institute y permite la reproducción de partes de su publicación para uso académico. La información de COBIT obtenida trata sobre los dominios, procesos, objetivos de control, objetivos de negocio, metas, métricas, acciones y modelos de madurez. Hay que hacer una observación y es que los objetivos TI aparecen en la aplicación sin información en la ventana de la solución y en otras ventanas de la administración debido a que no existe en el documento COBIT información explícita acerca de ellos. En cambio sí existe información en COBIT acerca de los objetivos de negocio (cuya información está introducida en la base de datos) que a su vez aparecen vacíos en la solución. Esto es así porque los objetivos de negocio están relacionados con los objetivos TI, pero al no tener datos sobre los objetivos TI, con el actual diseño de la base de datos, los objetivos de negocio no tienen ninguna relación establecida y es por ello que no aparecen tampoco en la solución. Sin embargo se ha optado por dejar ambos campos (objetivo TI y objetivo de negocio) y permitir la introducción de datos sobre ambos y sus relaciones para que en un futuro pueda existir esta información.

Por otro lado, las evaluaciones, utilizadas en la rama de la realización de evaluaciones y en administración, para obtener la madurez de la empresa, conocidas como assessments, provienen de una muestra de assessments disponible gratuitamente en <http://itgovernance.com> de IT Governance Network que a su vez se han basado en COBIT y proporcionan evaluaciones para los procesos PO1, PO9, PO10, AI6, DS5, DS11 y ME1 de COBIT.

Por último, los datos acerca de los problemas existentes en la base de datos, usados en la rama de consulta por problema y en administración, provienen de fuente propia y de dos páginas web:

- <http://www.securityprocedure.com/21-common-problems-within-iti-incident-management-implementation>
- <http://www.bestpricecomputers.ltd.uk/freehelp/common.htm>

4. Análisis

4.1 Análisis de Requisitos

En este apartado se procede a describir los requisitos de software del sistema. Estos requisitos han de definir con exactitud qué es lo hace el software desarrollado en este proyecto.

4.1.1 Usuarios y roles

En la aplicación existen dos tipos de perfiles de usuarios con roles perfectamente diferenciados:

- **Usuario:** Este tipo de perfil de usuario permite la interacción con el programa sin la modificación de los datos contenidos en la BBDD. Sí se pueden cambiar las preferencias visuales de la interfaz del programa. En concreto tiene acceso a las siguientes funciones:
 - **Metodología *COBIT 4.1*:**
 - Búsqueda de problemas y solución de los mismos mediante la metodología ***COBIT 4.1***.
 - Realización de tests de evaluación.
 - Consulta de procesos ***COBIT 4.1*** a través de dominios ***COBIT 4.1***.
 - **Opciones de la interfaz gráfica:**
(Contenidas en el fichero *ConfiguracionParametrosSW.txt*).
 - Modificación del idioma de la interfaz gráfica del programa – disponible en inglés y en español (este último no está implementado de forma completa en esta versión)-.
 - Modificación del número de problemas mostrados por ventana al realizar la búsqueda de problemas.
 - Modificación del número de evaluaciones mostradas por ventana al realizar un test de evaluación.
 - Modificación del tamaño de las letras de la barra de menú superior –en puntos-.
 - Modificación del color de la barra de menú superior –en formato RGB-.
 - Aplicar los valores por defecto.
 - **Guardar una solución:**
 - En formato txt, PDF, y HTML.

- **Impresión de una solución:** como demostración, la primera página de una solución.
- **Ayuda:**
 - Manual del usuario en idioma español o inglés (no disponible este último en esta versión) –en formato PDF-.
 - Información sobre el programa.
- **Administrador:** Este tipo de perfil de usuario permite la interacción con el programa y la modificación de los datos contenidos en la Base de Datos (BBDD), así como de dos de los tres ficheros de configuración que maneja el software. Se requiere de una contraseña para entrar en el modo administrador. En concreto el perfil tiene acceso a las siguientes funciones, además de las del perfil *Usuario*:
 - **Opciones de la BBDD:**
(*Contenidas y cifradas en el ficheroConfiguracionBaseDatos.txt*).
 - Modificación del host de la BBDD.
 - Modificación del puerto de la BBDD.
 - Modificación de la cadena de la BBDD.
 - Modificación del usuario de la BBDD.
 - Modificación de la contraseña de la BBDD.
 - **Modificación de la contraseña del usuario administrador.**
(*Contenida y cifrada en el ficheroPasswordAdministrador.txt*).
 - **Modificación de la información relacionada con los problemas de la BBDD:**
 - Añadir información:
 - Añadir problemas.
 - Añadir palabras.
 - Añadir sinónimos.
 - Eliminar información:
 - Eliminar problemas.
 - Eliminar palabras.
 - Eliminar sinónimos.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los problemas.
 - Modificar los nombres de las palabras.
 - Modificar los nombres de los sinónimos.
 - Relacionar información:
 - Relacionar/des-relacionar problemas con procesos **COBIT 4.1**.

- Relacionar/des-relacionar palabras con problemas.
 - Relacionar/des-relacionar palabras con sinónimos.
- **Modificación de la información relacionada con los tests de evaluación de la BBDD:**
 - Añadir información:
 - Añadir test de evaluación.
 - Eliminar información:
 - Eliminar test de evaluación.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los tests de evaluación.
 - Relacionar información:
 - Relacionar/des-relacionar tests de evaluación con procesos **COBIT 4.1**.
- **Modificación de la información relacionada con COBIT 4.1:**
 - Añadir información:
 - Añadir dominios.
 - Añadir procesos.
 - Añadir objetivos de control.
 - Añadir objetivos TI.
 - Añadir objetivos de negocio.
 - Añadir metas.
 - Añadir métricas.
 - Añadir acciones.
 - Añadir modelos de madurez.
 - Eliminar información:
 - Eliminar dominios.
 - Eliminar procesos.
 - Eliminar objetivos de control.
 - Eliminar objetivos TI.
 - Eliminar objetivos de negocio.
 - Eliminar metas.
 - Eliminar métricas.
 - Eliminar acciones.
 - Eliminar modelos de madurez.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los dominios.
 - Modificar los nombres de los procesos.

- Modificar los nombres de los objetivos de control.
 - Modificar los nombres de los objetivos TI.
 - Modificar los nombres de los objetivos de negocio.
 - Modificar los nombres de las metas.
 - Modificar los nombres de las métricas.
 - Modificar los nombres de las acciones.
 - Modificar los nombres de los modelos de madurez.
- Relacionar información:
 - Relacionar/des-relacionar dominios con procesos.
 - Relacionar/des-relacionar procesos con objetivos de control.
 - Relacionar/des-relacionar objetivos de control con objetivos TI.
 - Relacionar/des-relacionar objetivos TI con objetivos de negocio.
 - Relacionar/des-relacionar metas con métricas.
 - Relacionar/des-relacionar métricas con objetivos de control.
 - Relacionar/des-relacionar acciones con objetivos de control.
 - Relacionar/des-relacionar modelos de madurez con procesos.

4.2 Especificación de Requisitos

A continuación se pasa a especificar la forma en que serán representados los requisitos, así como la clasificación por tipos de estos:

Id			
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad			
Descripción			

Tabla 1. Plantilla genérica de requisitos

Id: Identifica unívocamente cada requisito de usuario facilitando la trazabilidad. El formato que seguirá el identificador será el siguiente:

RS-[TIPO][XXX]

Donde:

- **RS:** Término para designar Requisitos de Software.
- **TIPO:** Tipo del requisito que, en nuestro caso, podrá tomar los valores:
 - **F:** Designa al requisito el tipo de *Funcional*. Este tipo de requisitos simbolizan lo que tiene que realizar el software.
 - **R:** Designa al requisito el tipo de *Rendimiento*. Este tipo de requisitos especifican valores numéricos para variables de rendimiento como pueden ser tasas de transferencia, velocidad de proceso, etc.
 - **I:** Designa al requisito el tipo de *Interfaz*. Este tipo de requisitos especifican hardware o software con el que la aplicación interactuará.
 - **RE:** Designa al requisito el tipo de *Recursos*. Este tipo de requisitos detallan los límites superiores de los recursos físicos a utilizar por el sistema, tales como memoria, espacio de disco, etc.
 - **C:** Designa al requisito el tipo de *Comprobación*. Este tipo de requisitos resaltan como debe la aplicación verificar los datos de entrada y salida.
 - **D:** Designa al requisito el tipo de *Documentación*. Este tipo de requisitos establecen la manera en la que realizar la documentación del proyecto.

- **S:** Designa al requisito el tipo de *Seguridad*. Este tipo de requisitos determinan las pautas para asegurar al sistema contra amenazas que comprometan la confidencialidad, integridad y disponibilidad de los datos.
 - **DA:** Designa al requisito el tipo de *Daño*. Este tipo de requisitos detallan cualquier requisito que pudiera surgir por el fracaso del software, para reducir el posible daño.
- **XXX:** Número de requisito que se otorga de forma secuencial y empezando por 1.

Ejemplos:

- **RS-F001:** Requisito de Software Funcional número 1.
- **RS-S001:** Requisito de Software de Seguridad número 1.

Prioridad: Cada requisito contendrá una medida de la prioridad que tiene, para que el desarrollador pueda hacer una planificación acorde a las necesidades impuestas. Los valores que puede tomar son:

- **Alta:** Son los primeros en tener que ser desarrollados.
- **Media:** Importante no dejarlos para el final del desarrollo.
- **Baja:** Pueden ser desarrollados en cualquier momento.

Fuente: Origen de cada requisito.

Necesidad: Representa el grado de importancia que el equipo de desarrollo da a ciertos aspectos. Los valores que puede tomar son:

- **Esencial:** Son innegociables.
- **Deseable:** Su inclusión puede ser negociada.
- **Opcional:** Aspectos poco importantes y poco relevantes.

Claridad: Un requisito de software únicamente ha de tener una interpretación posible. Implica la no ambigüedad.

- **Alta:** El requisito es claro.
- **Media:** El requisito puede dar lugar a confusiones.
- **Baja:** El requisito es ambiguo.

Verificabilidad: Se debe poder comprobar que cada requisito está aplicado en el software. Puede tomar los siguientes valores:

- **Alta:** Fácilmente demostrable.
- **Media:** Demostrable sin cierta dificultad.
- **Baja:** Difícilmente demostrable.

Estabilidad: Indica la probabilidad que tiene un requisito en ser modificado posteriormente. Toma los siguientes valores:

- **Estable:** Señala que el requisito no será modificado en el ciclo de vida del software.
- **Inestable:** Señala que en posteriores fases, puede llevarse a cabo la modificación del requisito debido a un feedback producido sobre el mismo requisito.

Descripción: Explicación clara y concisa de lo que pretende cada requisito.

Una vez se ha descrito la manera en la que representará cada requisito de software y lo que significa cada concepto, se procede a listar la totalidad de los requisitos de software del sistema:

4.2.1 Requisitos funcionales

Id	RS-F001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El usuario de la aplicación podrá consultar cualquier dominio y proceso de la metodología COBIT 4.1 .		

Tabla 2. Requisito software RS-F001

Id	RS-F002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El usuario de la aplicación puede cambiar preferencias de la interfaz gráfica.		

Tabla 3. Requisito software RS-F002

Id	RS-F003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El usuario de la aplicación podrá consultar la ayuda del software.		

Tabla 4. Requisito software RS-F003

Id	RS-F004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El administrador de la aplicación podrá cambiar la contraseña de acceso a la administración.		

Tabla 5. Requisito software RS-F004

Id	RS-F005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El administrador de la aplicación podrá modificar los parámetros de conexión con la Base de Datos.		

Tabla 6. Requisito software RS-F005

Id	RS-F006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El administrador de la aplicación podrá modificar la información relacionada con la consulta de problemas de la rama del usuario, pudiendo efectuar cambios en los nombres y relaciones de los problemas, palabras, y sinónimos, incluidos en la Base de Datos.		

Tabla 7. Requisito software RS-F006

Id	RS-F007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El administrador de la aplicación podrá modificar la información relacionada con la realización de los tests de evaluación de la rama del usuario, pudiendo efectuar cambios en los nombres y relaciones de los tests de evaluación incluidos en la Base de Datos.		

Tabla 8. Requisito software RS-F007

Id	RS-F008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El administrador de la aplicación podrá modificar la información relacionada con la metodología COBIT 4.1 de la rama del usuario, pudiendo efectuar cambios en los nombres y relaciones de la información de COBIT 4.1 incluida en la Base de Datos.		

Tabla 9. Requisito software RS-F008

Id	RS-F009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	<p>La aplicación deberá mostrar a los usuarios la solución mediante la metodología COBIT 4.1, a través de cada una de las ramas por las que puede hacer una consulta el usuario a la aplicación:</p> <ul style="list-style-type: none"> - Búsqueda de problemas relacionados con TI. - Realización de tests de evaluación. - Consulta a través de dominios y procesos de COBIT 4.1. 		

Tabla 10. Requisito software RS-F009

Id	RS-F010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El usuario podrá imprimir la primera hoja de la solución mostrada en pantalla, a modo de demostración.		

Tabla 11. Requisito software RS-F010

Id	RS-F011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	<p>El usuario podrá guardar la solución mostrada en pantalla en los siguientes formatos:</p> <ul style="list-style-type: none"> - txt. - PDF. - HTML. 		

Tabla 12. Requisito software RS-F011

4.2.2 Requisitos de rendimiento

Id	RS-R001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	El administrador de la aplicación será el encargado de obtener la configuración óptima entre la base de datos y la aplicación para conseguir el mejor rendimiento.		

Tabla 13. Requisito de rendimiento RS-R001

4.2.3 Requisitos de interfaz

Id	RS-I001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	<p>La aplicación dispondrá de una interfaz bien definida para que sea lo más usable posible a la vez que eficiente de cara a mostrar todos los datos por pantalla:</p> <ul style="list-style-type: none"> • En la parte superior habrá una barra de título. • Debajo de la barra de título, habrá una barra de herramientas/menú superior. • Debajo de la barra de herramientas/menú superior habrá un panel principal donde se mostrará toda la información relevante de la aplicación. 		

Tabla 14. Requisito de interfaz RS-I001

Id	RS-I002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	<p>El usuario podrá modificar los siguientes parámetros de la interfaz:</p> <ul style="list-style-type: none"> • Idioma español (en versión de prueba en esta versión de la aplicación), o inglés. • Número de problemas mostrados en el panel principal al realizar una consulta por el usuario desde la rama de consulta de problemas. • Número de tests de evaluación mostrados en el panel principal al realizar los tests de evaluación. • Tamaño en puntos de la letra de la barra de herramientas/menú superior. • Color de fondo de la barra de herramientas/menú superior, en formato RGB. • Aplicar los valores por defecto de la interfaz. 		

Tabla 15. Requisito de interfaz RS-I002

4.2.4 Requisitos de recursos

Id	RS-RE001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	<p>Para que la aplicación funcione correctamente, se ha de disponer del siguiente hardware y software instalado:</p> <ul style="list-style-type: none"> • PC compatible con Microsoft Windows. • Sistema Operativo compatible con Microsoft Windows, tanto en versión de 32 bits como de 64 bits. • Bases de Datos compatible con Oracle. • Java Runtime Environment. • Compresor/Descompresor de ficheros en formato Zip. • Visor de documentos en PDF. • Visor de texto plano. • Navegador de Internet compatible con HTML. 		

Tabla 16. Requisito de recurso RS-RE001

4.2.5 Requisitos de comprobación

Id	RS-C001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	<p>Comprobar que la aplicación deniega el acceso a la administración cuando la contraseña introducida por el administrador no es correcta.</p>		

Tabla 17. Requisito de comprobación RS-C001

Id	RS-C002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	Comprobar que la aplicación permite el acceso a la administración cuando la contraseña introducida por el administrador es correcta.		

Tabla 18. Requisito de comprobación RS-C002

Id	RS-C003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	Comprobar que la aplicación permite la conexión con la Base de Datos de Oracle cuando los parámetros de conexión a la Base de Datos están bien configurados.		

Tabla 19. Requisito de comprobación RS-C003

4.2.6 Requisitos de documentación

Id	RS-D001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	Se facilitará a los usuarios un manual del usuario en idioma español especificando todas las funcionalidades de la aplicación.		

Tabla 20. Requisito de documentación RS-D001

4.2.7 Requisitos de seguridad

Id	RS-S001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	Los parámetros de conexión con la Base de Datos, y la contraseña del administrador, estarán cifrados mediante un algoritmo de cifrado, y se almacenarán en ficheros de texto plano para su conservación en un disco duro.		

Tabla 21. Requisito de seguridad RS-S001

4.2.8 Requisitos de daño

Id	RS-DA001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estable		
Descripción	Se garantizará que la aplicación será un sistema robusto frente a fallos, es decir, soportará que cualquier acción del usuario y del administrador que pueda producir un error, y se identificará inmediatamente por pantalla.		

Tabla 22. Requisito de daño RS-DA001

4.3 Diagrama de Casos de Uso

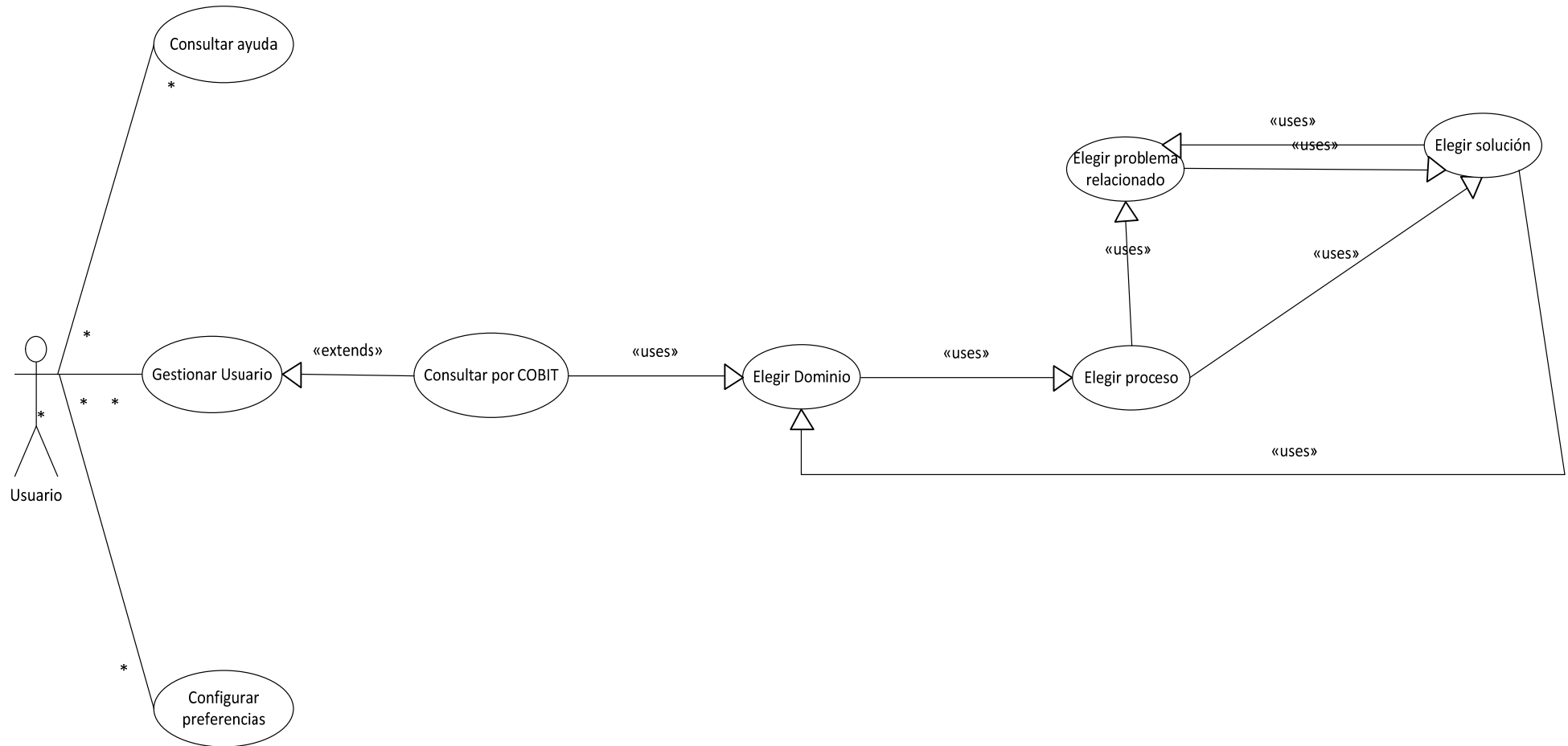


Ilustración 12. Casos de Uso de Usuario

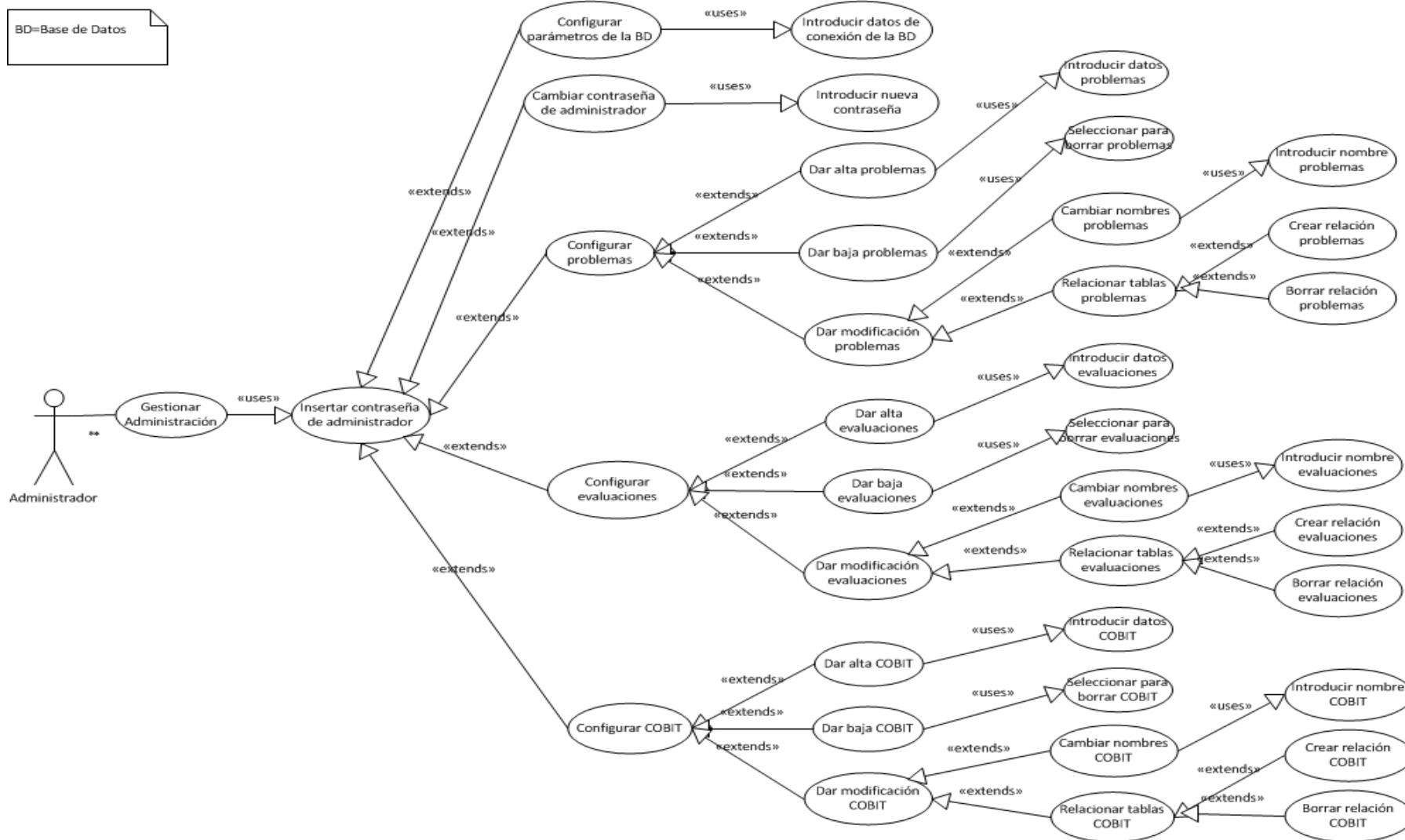


Ilustración 13. Casos de Uso de Administrador

4.3.1 Descripción textual de Casos de Uso del Usuario

Nombre	Gestionar Usuario.
Actores	Usuario.
Objetivo	El usuario elige el perfil usuario de la aplicación.
Precondiciones	El usuario se encuentra en la ventana principal.
Postcondiciones	El usuario podrá realizar consultas.
Escenario básico	1. Elegir perfil usuario.

Tabla 23. Caso de Uso Gestionar Usuario

Nombre	Consultar por COBIT.
Actores	Usuario.
Objetivo	El usuario desea realizar una consulta teniendo conocimientos sobre COBIT.
Precondiciones	El usuario eligió el perfil usuario.
Postcondiciones	El usuario podrá elegir un dominio de COBIT.
Escenario básico	1. Elegir perfil usuario. 2. Elegir consultar por COBIT.

Tabla 24. Caso de Uso Consultar por COBIT

Nombre	Elegir dominio.
Actores	Usuario.
Objetivo	El usuario elige un dominio de COBIT
Precondiciones	El usuario eligió el perfil usuario y ha elegido consultar por COBIT.
Postcondiciones	El usuario podrá elegir un proceso asociado al dominio elegido.
Escenario básico	1. Elegir perfil usuario. 2. Elegir consultar por COBIT. 3. Elegir dominio.

Tabla 25. Caso de Uso Elegir dominio

Nombre	Elegir proceso.
Actores	Usuario.
Objetivo	El usuario elige un proceso relacionado con el dominio elegido.
Precondiciones	El usuario eligió el perfil usuario, ha elegido consultar por COBIT y ha elegido un dominio.
Postcondiciones	El usuario podrá elegir un problema relacionado con el suyo o elegir una solución.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil usuario. 2. Elegir consultar por COBIT. 3. Elegir dominio. 4. Elegir proceso. <ol style="list-style-type: none"> 4.1. Elegir problema relacionado. <ol style="list-style-type: none"> 4.1.1. Elegir solución. 4.2. Elegir solución.

Tabla 26. Caso de Uso Elegir proceso

Nombre	Elegir problema relacionado.
Actores	Usuario.
Objetivo	El usuario elige un problema relacionado con la consulta por evaluación o con la consulta por COBIT.
Precondiciones	El usuario eligió el perfil usuario, ha elegido consultar por evaluación, ha introducido respuestas y ha elegido un proceso con madurez. O bien el usuario eligió el perfil usuario, ha elegido consultar por COBIT, ha elegido un dominio y ha elegido un proceso.
Postcondiciones	El usuario podrá elegir una solución.
Escenario básico	<p>Por la rama de consulta por evaluación:</p> <ol style="list-style-type: none"> 1. Elegir perfil usuario. 2. Elegir consultar por evaluación. 3. Elegir respuestas. 4. Elegir proceso con madurez. 5. Elegir problema relacionado. <p>O bien por la rama de consulta por COBIT:</p> <ol style="list-style-type: none"> 1. Elegir perfil usuario. 2. Elegir consultar por COBIT. 3. Elegir dominio. 4. Elegir proceso. 5. Elegir problema relacionado.

Tabla 27. Caso de Uso Elegir problema relacionado

Nombre	Elegir solución.
Actores	Usuario.
Objetivo	El usuario elige una solución.
Precondiciones	El usuario eligió el perfil usuario, ha elegido consultar por problema, ha introducido su problema y ha elegido un problema. O bien el usuario eligió el perfil usuario, ha elegido consultar por evaluación, ha introducido respuestas y ha elegido un proceso con madurez. O bien el usuario eligió el perfil usuario, ha elegido consultar por COBIT, ha elegido un dominio y ha elegido un proceso.
Postcondiciones	El usuario podrá elegir otra solución o repetir por la rama de consulta elegida en la precondición.
Escenario básico	<p>Por la rama consulta por problema:</p> <ol style="list-style-type: none"> 1. Elegir perfil usuario. 2. Elegir consultar por problema. 3. Introducir problema. 4. Elegir problema. <p>Por la rama consulta por evaluación:</p> <ol style="list-style-type: none"> 1. Elegir perfil usuario. 2. Elegir consultar por evaluación. 3. Elegir respuestas. 4. Elegir proceso con madurez. <ol style="list-style-type: none"> 4.1. Elegir problema relacionado. <ol style="list-style-type: none"> 4.1.1. Elegir solución. 4.2. Elegir solución. <p>Por la rama consulta por COBIT:</p> <ol style="list-style-type: none"> 1. Elegir perfil usuario. 2. Elegir consultar por COBIT. 3. Elegir dominio. 4. Elegir proceso. <ol style="list-style-type: none"> 4.1. Elegir problema relacionado. <ol style="list-style-type: none"> 4.1.1. Elegir solución. 4.2. Elegir solución.

Tabla 28. Caso de Uso Elegir solución

Nombre	Consultar ayuda.
Actores	Usuario.
Objetivo	El usuario desea utilizar la ayuda del manual del usuario.
Precondiciones	El usuario se encuentra en cualquier ventana de la aplicación.
Postcondiciones	El usuario obtendrá un fichero en formato PDF donde consultar la ayuda.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar el menú ayuda (Help). 2. Seleccionar el ítem manual de usuario (User guide).

Tabla 29. Caso de Uso Consultar ayuda

Nombre	Configurar preferencias.
Actores	Usuario.
Objetivo	El usuario desea configurar preferencias de interfaz gráfica, de idioma u otros aspectos.
Precondiciones	El usuario se encuentra en cualquier ventana de la aplicación.
Postcondiciones	El usuario cambiará el aspecto, el idioma u otras funcionalidades.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar el menú opciones (Options). <ol style="list-style-type: none"> 1.1. Seleccionar el ítem del idioma (Language). 1.2. Seleccionar el ítem de preferencias (Preferences).

Tabla 30. Caso de Uso Configurar preferencias

4.3.2 Descripción textual de Casos de Uso del Administrador

Nombre	Gestionar Administración.
Actores	Administrador.
Objetivo	El administrador elige el perfil administrador de la aplicación.
Precondiciones	El administrador se encuentra en la ventana principal.
Postcondiciones	El administrador se identificará como administrador.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador.

Tabla 31. Caso de Uso Gestionar Administración

Nombre	Insertar contraseña de administrador.
Actores	Administrador.
Objetivo	El administrador se valida como administrador.
Precondiciones	Perfil administrador está identificándose en la ventana para introducir contraseña.
Postcondiciones	Administrador identificado como administrador.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. <ol style="list-style-type: none"> 2.1. Si es correcto, administrador validado. 2.2. Si no, vuelve al punto 2.

Tabla 32. Caso de Uso Insertar contraseña de administrador

Nombre	Configurar parámetros de la BD.
Actores	Administrador.
Objetivo	El administrador desea cambiar los parámetros de la conexión con la base de datos.
Precondiciones	El administrador se ha logueado correctamente.
Postcondiciones	El administrador puede realizar cambios de la conexión en la base de datos.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir cambiar parámetros de la Base de Datos.

Tabla 33. Caso de Uso Configurar parámetros de la BD

Nombre	Introducir datos de conexión de la base de datos.
Actores	Administrador.
Objetivo	El administrador modifica los datos de la conexión con la base de datos.
Precondiciones	El administrador se ha logueado correctamente y ha elegido cambiar datos de conexión de la base de datos.
Postcondiciones	El administrador modifica datos de la conexión con la base de datos.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir cambiar parámetros de la Base de Datos. 5. Introducir datos de la conexión.

Tabla 34. Caso de Uso Introducir datos de conexión de la base de datos

Nombre	Cambiar contraseña de administrador.
Actores	Administrador.
Objetivo	El administrador desea cambiar la contraseña de identificación del administrador.
Precondiciones	El administrador se ha logueado correctamente.
Postcondiciones	El administrador tiene una nueva contraseña.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir cambiar contraseña de administrador.

Tabla 35. Cambiar contraseña de administrador

Nombre	Introducir nueva contraseña.
Actores	Administrador.
Objetivo	El administrador cambia la contraseña de identificación del administrador.
Precondiciones	El administrador se ha logueado correctamente y ha elegido cambiar la contraseña del administrador.
Postcondiciones	El administrador tiene una nueva contraseña.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir cambiar contraseña de administrador. 5. Introducir nueva contraseña.

Tabla 36. Caso de Uso Introducir nueva contraseña

Nombre	Configurar problemas.
Actores	Administrador.
Objetivo	El administrador desea configurar los problemas.
Precondiciones	El administrador se ha logueado correctamente y desea cambiar la configuración de los problemas.
Postcondiciones	El administrador podrá elegir entre altas, bajas y modificaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas.

Tabla 37. Configurar problemas

Nombre	Dar alta problemas.
Actores	Administrador.
Objetivo	El administrador desea dar de alta nuevos problemas.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar problemas.
Postcondiciones	El administrador podrá introducir nuevos problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar de alta problemas.

Tabla 38. Caso de Uso Dar alta problemas

Nombre	Introducir datos problemas.
Actores	Administrador.
Objetivo	El administrador va a introducir nuevos problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas y ha elegido dar de alta.
Postcondiciones	El administrador añadirá nuevos problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar de alta problemas. 6. Introducir nuevas altas.

Tabla 39. Caso de Uso Introducir datos problemas

Nombre	Dar baja problemas.
Actores	Administrador.
Objetivo	El administrador desea dar de baja a problemas.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar problemas.
Postcondiciones	El administrador podrá borrar problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar de baja problemas.

Tabla 40. Caso de Uso Dar baja problemas

Nombre	Seleccionar para borrar problemas.
Actores	Administrador.
Objetivo	El administrador va a borrar problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas y ha elegido dar de baja.
Postcondiciones	El administrador borrará problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Seleccionar problemas para dar de baja. 6. Borrar problemas.

Tabla 41. Caso de Uso Seleccionar para borrar problemas

Nombre	Dar modificación problemas.
Actores	Administrador.
Objetivo	El administrador desea modificar problemas.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar problemas.
Postcondiciones	El administrador podrá elegir entre cambiar nombres o relacionar tablas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar modificación problemas.

Tabla 42. Caso de Uso Dar modificación problemas

Nombre	Cambiar nombres problemas.
Actores	Administrador.
Objetivo	El administrador desea cambiar nombres de los problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas y ha elegido dar modificación a problemas.
Postcondiciones	El administrador podrá introducir nuevos nombres para problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar modificación problemas. 6. Elegir cambiar nombres de problemas.

Tabla 43. Caso de Uso Cambiar nombres problemas

Nombre	Introducir nombres problemas.
Actores	Administrador.
Objetivo	El administrador cambiará nombres de los problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas, ha elegido dar modificación a problemas y ha elegido cambiar nombres.
Postcondiciones	El administrador introducirá nuevos nombres para problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar modificación problemas. 6. Elegir cambiar nombres de problemas. 7. Introducir nuevos nombres de problemas.

Tabla 44. Introducir nombres problemas

Nombre	Relacionar tablas problemas.
Actores	Administrador.
Objetivo	El administrador desea relacionar tablas de los problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas y ha elegido dar modificación a problemas.
Postcondiciones	El administrador podrá crear o borrar relaciones para problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar modificación problemas. 6. Elegir relacionar tablas de problemas.

Tabla 45. Caso de Uso Relacionar tablas problemas

Nombre	Crear relación problemas.
Actores	Administrador.
Objetivo	El administrador creará una relación de tablas de los problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas, ha elegido dar modificación a problemas y ha elegido crear relación de tablas.
Postcondiciones	El administrador creará relaciones para problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar modificación problemas. 6. Elegir relacionar tablas de problemas.

Tabla 46. Caso de Uso Crear relación problemas

Nombre	Borrar relación problemas.
Actores	Administrador.
Objetivo	El administrador borrará una relación de tablas de los problemas.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar problemas, ha elegido dar modificación a problemas y ha elegido borrar relación de tablas.
Postcondiciones	El administrador borrará relaciones para problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar problemas. 5. Elegir dar modificación problemas. 6. Elegir borrar relaciones de tablas de problemas.

Tabla 47. Caso de Uso Borrar relación problemas

Nombre	Configurar evaluaciones.
Actores	Administrador.
Objetivo	El administrador desea configurar las evaluaciones.
Precondiciones	El administrador se ha logueado correctamente y desea cambiar la configuración de las evaluaciones.
Postcondiciones	El administrador podrá elegir entre altas, bajas y modificaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones.

Tabla 48. Caso de Uso Configurar evaluaciones

Nombre	Dar alta evaluaciones.
Actores	Administrador.
Objetivo	El administrador desea dar de alta nuevas evaluaciones.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar evaluaciones.
Postcondiciones	El administrador podrá introducir nuevas evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar de alta evaluaciones.

Tabla 49. Caso de Uso Dar alta evaluaciones

Nombre	Introducir datos evaluaciones.
Actores	Administrador.
Objetivo	El administrador va a introducir nuevas evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones y ha elegido dar de alta.
Postcondiciones	El administrador añadirá nuevos problemas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar de alta evaluaciones. 6. Introducir nuevas altas.

Tabla 50. Caso de Uso Introducir datos evaluaciones

Nombre	Dar baja evaluaciones.
Actores	Administrador.
Objetivo	El administrador desea dar de baja a evaluaciones.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar evaluaciones.
Postcondiciones	El administrador podrá borrar evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar de baja evaluaciones.

Tabla 51. Caso de Uso Dar baja evaluaciones

Nombre	Seleccionar para borrar evaluaciones.
Actores	Administrador.
Objetivo	El administrador va a borrar evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones y ha elegido dar de baja.
Postcondiciones	El administrador borrará evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 7. Elegir perfil administrador. 8. Introducir contraseña. 9. Administrador validado. 10. Elegir configurar evaluaciones. 11. Seleccionar evaluaciones para dar de baja. 12. Borrar evaluaciones.

Tabla 52. Caso de Uso Seleccionar para borrar evaluaciones

Nombre	Dar modificación evaluaciones.
Actores	Administrador.
Objetivo	El administrador desea modificar evaluaciones.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar evaluaciones.
Postcondiciones	El administrador podrá elegir entre cambiar nombres o relacionar tablas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar modificación evaluaciones.

Tabla 53. Caso de Uso Dar modificación evaluaciones

Nombre	Cambiar nombres evaluaciones.
Actores	Administrador.
Objetivo	El administrador desea cambiar nombres de las evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones y ha elegido dar modificación a evaluaciones.
Postcondiciones	El administrador podrá introducir nuevos nombres para evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar modificación evaluaciones. 6. Elegir cambiar nombres de evaluaciones.

Tabla 54. Caso de Uso Cambiar nombres evaluaciones

Nombre	Introducir nombres evaluaciones.
Actores	Administrador.
Objetivo	El administrador cambiará nombres de las evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones, ha elegido dar modificación a evaluaciones y ha elegido cambiar nombres.
Postcondiciones	El administrador introducirá nuevos nombres para evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar modificación evaluaciones. 6. Elegir cambiar nombres de evaluaciones. 7. Introducir nuevos nombres de evaluaciones.

Tabla 55. Caso de Uso Introducir nombres evaluaciones

Nombre	Relacionar tablas evaluaciones.
Actores	Administrador.
Objetivo	El administrador desea relacionar tablas de las evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones y ha elegido dar modificación a evaluaciones.
Postcondiciones	El administrador podrá crear o borrar relaciones para evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar modificación evaluaciones. 6. Elegir relacionar tablas de evaluaciones.

Tabla 56. Caso de Uso Relacionar tablas evaluaciones

Nombre	Crear relación evaluaciones.
Actores	Administrador.
Objetivo	El administrador creará una relación de tablas de las evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones, ha elegido dar modificación a evaluaciones y ha elegido crear relación de tablas.
Postcondiciones	El administrador creará relaciones para evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar modificación evaluaciones. 6. Elegir relacionar tablas de evaluaciones.

Tabla 57. Caso de Uso Crear relación evaluaciones

Nombre	Borrar relación evaluaciones.
Actores	Administrador.
Objetivo	El administrador borrará una relación de tablas de las evaluaciones.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar evaluaciones, ha elegido dar modificación a evaluaciones y ha elegido borrar relación de tablas.
Postcondiciones	El administrador borrará relaciones para evaluaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar evaluaciones. 5. Elegir dar modificación evaluaciones. 6. Elegir borrar relaciones de tablas de evaluaciones.

Tabla 58. Caso de Uso Borrar relación evaluaciones

Nombre	Configurar COBIT.
Actores	Administrador.
Objetivo	El administrador desea configurar COBIT.
Precondiciones	El administrador se ha logueado correctamente y desea cambiar la configuración de COBIT.
Postcondiciones	El administrador podrá elegir entre altas, bajas y modificaciones.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT.

Tabla 59. Caso de Uso Configurar COBIT

Nombre	Dar alta COBIT.
Actores	Administrador.
Objetivo	El administrador desea dar de alta nueva información de COBIT.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar COBIT.
Postcondiciones	El administrador podrá introducir nueva información de COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar de alta COBIT.

Tabla 60. Caso de Uso Dar alta COBIT

Nombre	Introducir datos COBIT.
Actores	Administrador.
Objetivo	El administrador va a introducir nueva información de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT y ha elegido dar de alta.
Postcondiciones	El administrador añadirá nueva información de COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar de alta COBIT. 6. Introducir nuevas altas.

Tabla 61. Caso de Uso Introducir datos COBIT

Nombre	Dar baja COBIT.
Actores	Administrador.
Objetivo	El administrador desea dar de baja a información de COBIT.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar COBIT.
Postcondiciones	El administrador podrá borrar información de COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar de baja COBIT.

Tabla 62. Caso de Uso Dar baja COBIT

Nombre	Seleccionar para borrar COBIT.
Actores	Administrador.
Objetivo	El administrador va a borrar información de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT y ha elegido dar de baja.
Postcondiciones	El administrador borrará información de COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Seleccionar información de COBIT para dar de baja. 6. Borrar información de COBIT.

Tabla 63. Caso de Uso Seleccionar para borrar COBIT

Nombre	Dar modificación COBIT.
Actores	Administrador.
Objetivo	El administrador desea modificar información de COBIT.
Precondiciones	El administrador se ha logueado correctamente y ha elegido configurar COBIT.
Postcondiciones	El administrador podrá elegir entre cambiar nombres o relacionar tablas.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar modificación COBIT.

Tabla 64. Caso de Uso Dar modificación COBIT

Nombre	Cambiar nombres COBIT.
Actores	Administrador.
Objetivo	El administrador desea cambiar nombres de la información de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT y ha elegido dar modificación a COBIT.
Postcondiciones	El administrador podrá introducir nuevos nombres para la información de COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar modificación COBIT. 6. Elegir cambiar nombres de COBIT.

Tabla 65. Caso de Uso Cambiar nombres COBIT

Nombre	Introducir nombres COBIT.
Actores	Administrador.
Objetivo	El administrador cambiará nombres de la información de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT, ha elegido dar modificación a COBIT y ha elegido cambiar nombres.
Postcondiciones	El administrador introducirá nuevos nombres para la información de COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar modificación COBIT. 6. Elegir cambiar nombres de COBIT. 7. Introducir nuevos nombres de COBIT.

Tabla 66. Caso de Uso Introducir nombres COBIT

Nombre	Relacionar tablas COBIT.
Actores	Administrador.
Objetivo	El administrador desea relacionar tablas de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT y ha elegido dar modificación a COBIT.
Postcondiciones	El administrador podrá crear o borrar relaciones para COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar modificación COBIT. 6. Elegir relacionar tablas de COBIT.

Tabla 67. Caso de Uso Relacionar tablas COBIT

Nombre	Crear relación COBIT.
Actores	Administrador.
Objetivo	El administrador creará una relación de tablas de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT, ha elegido dar modificación a COBIT y ha elegido crear relación de tablas.
Postcondiciones	El administrador creará relaciones para COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar modificación COBIT. 6. Elegir relacionar tablas de COBIT.

Tabla 68. Caso de Uso Crear relación COBIT

Nombre	Borrar relación COBIT.
Actores	Administrador.
Objetivo	El administrador borrará una relación de tablas de COBIT.
Precondiciones	El administrador se ha logueado correctamente, ha elegido configurar COBIT, ha elegido dar modificación a COBIT y ha elegido borrar relación de tablas.
Postcondiciones	El administrador borrará relaciones para COBIT.
Escenario básico	<ol style="list-style-type: none"> 1. Elegir perfil administrador. 2. Introducir contraseña. 3. Administrador validado. 4. Elegir configurar COBIT. 5. Elegir dar modificación COBIT. 6. Elegir borrar relaciones de tablas de COBIT.

Tabla 69. Caso de Uso Borrar relación COBIT

4.4 Diagramas de Secuencia

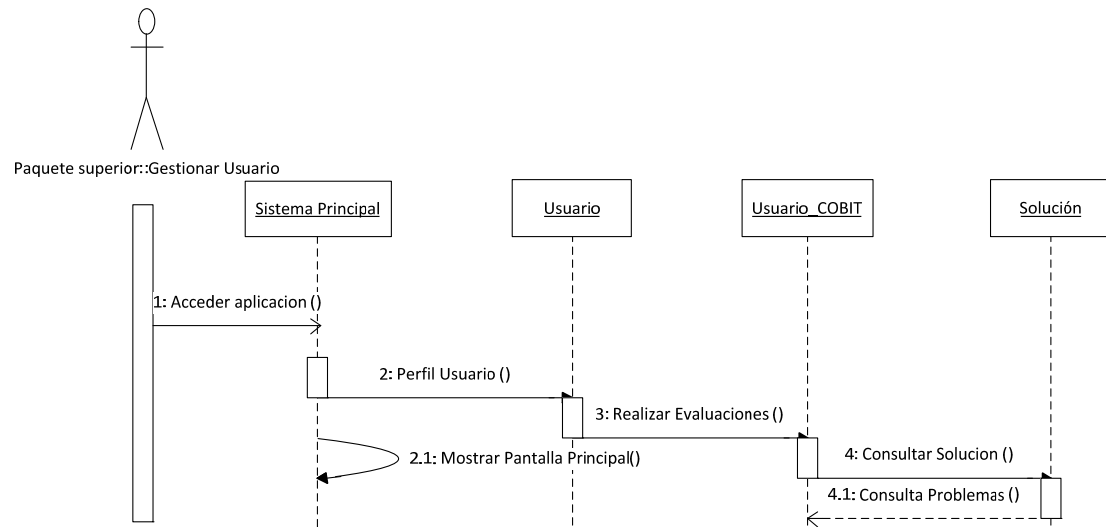


Ilustración 14. Diagrama de Secuencia Buscar por dominio y proceso de COBIT

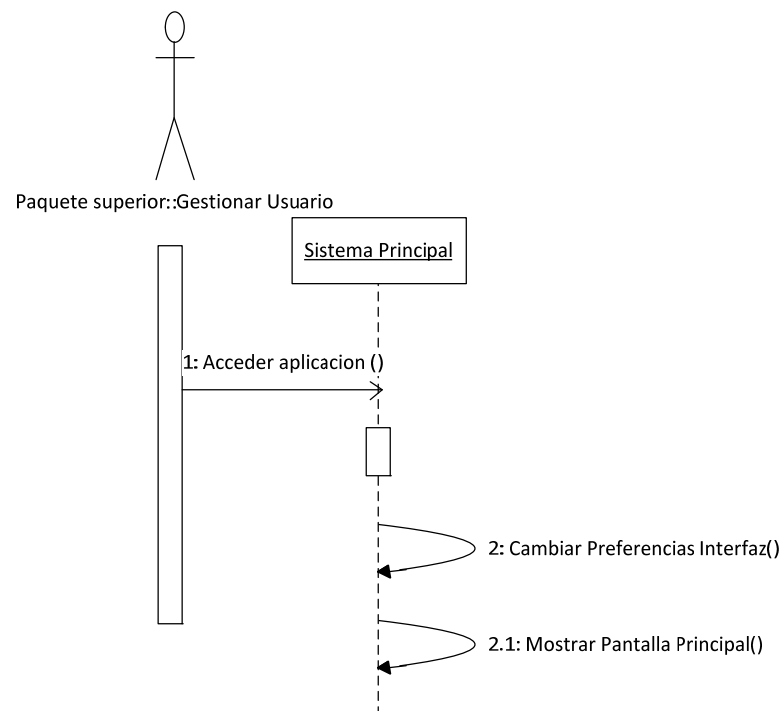


Ilustración 15. Diagrama de Secuencia Cambiar preferencias interfaz

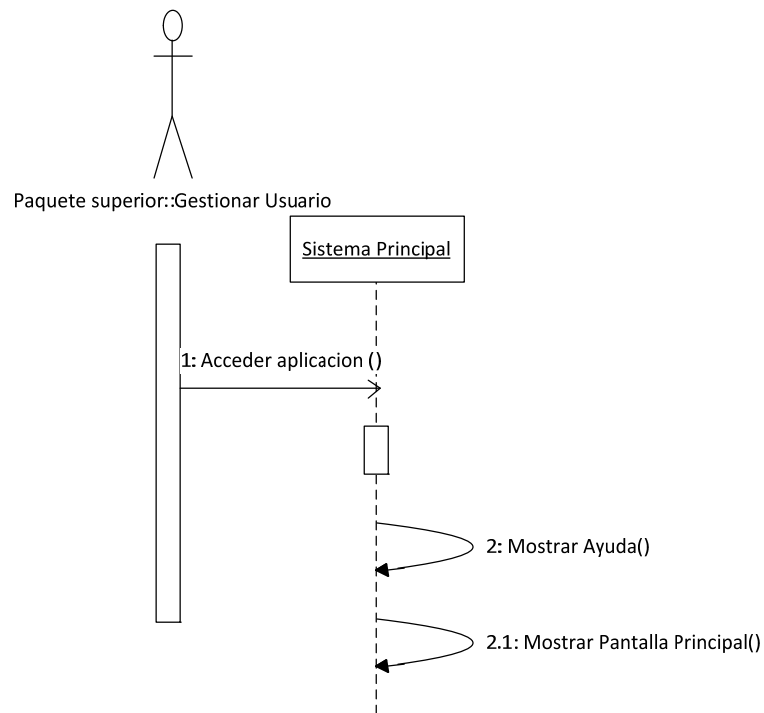


Ilustración 16. Diagrama de Secuencia Ayuda

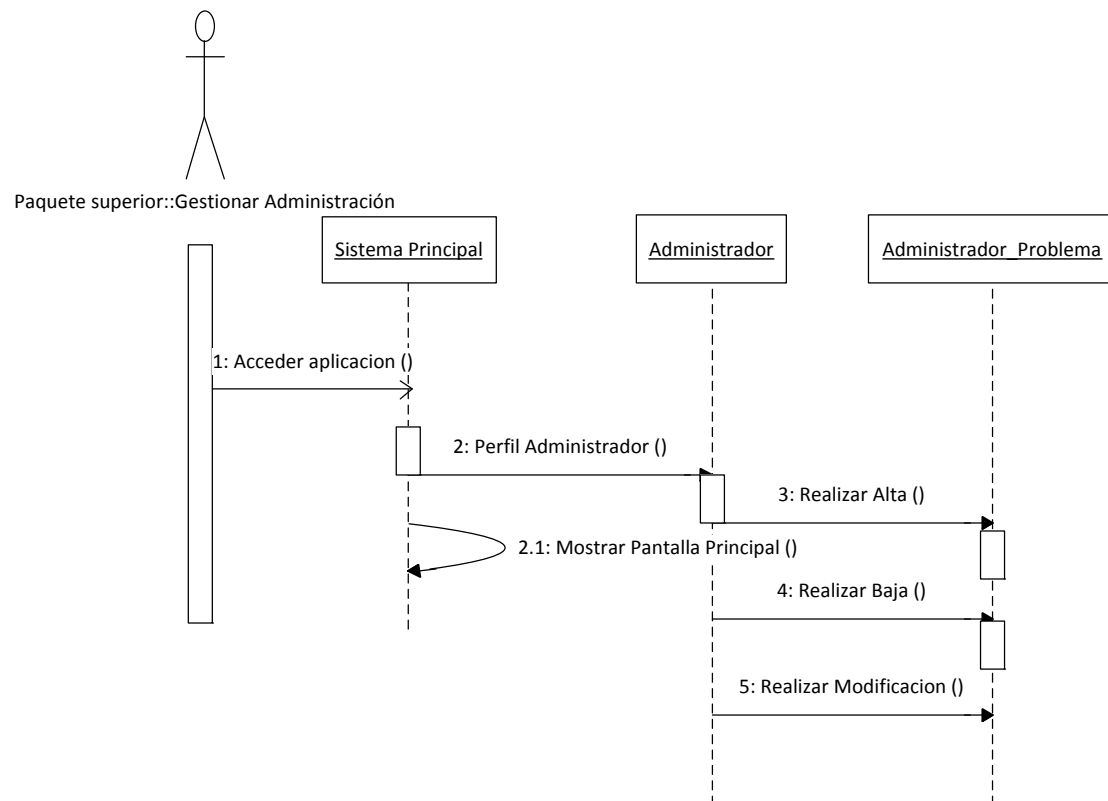


Ilustración 17. Diagrama de Secuencia Administración problemas

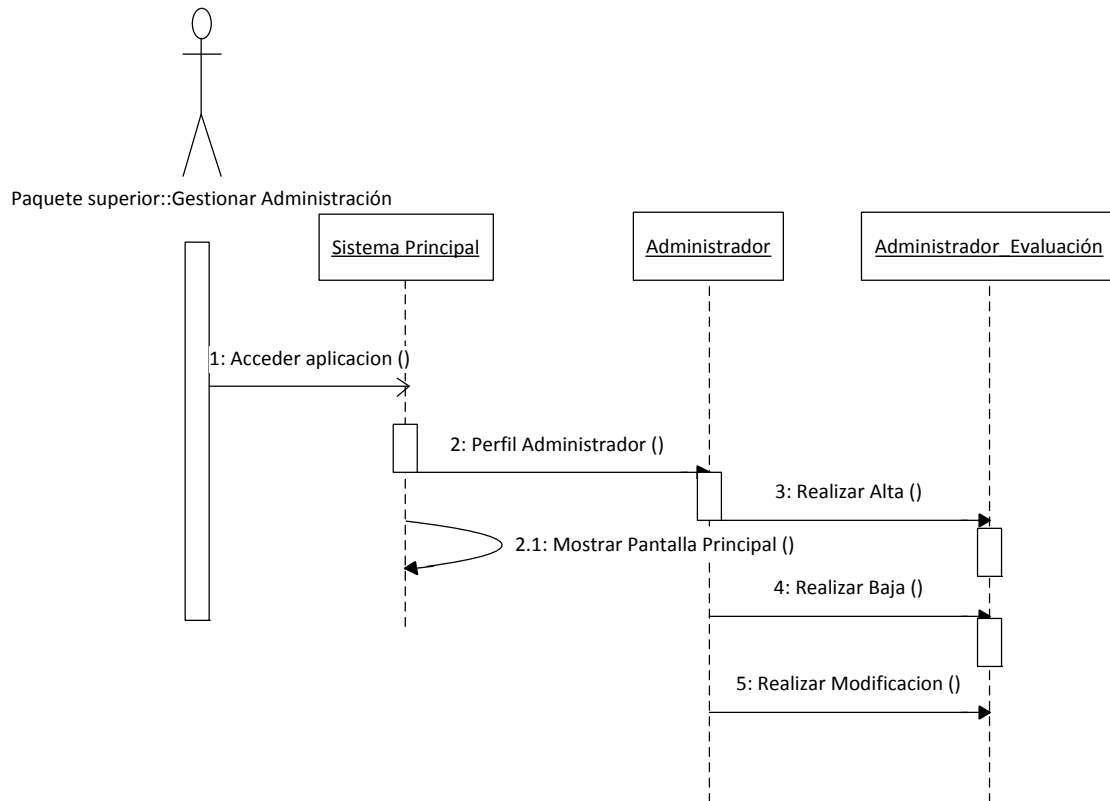


Ilustración 18. Diagrama de Secuencia Administración tests de evaluación

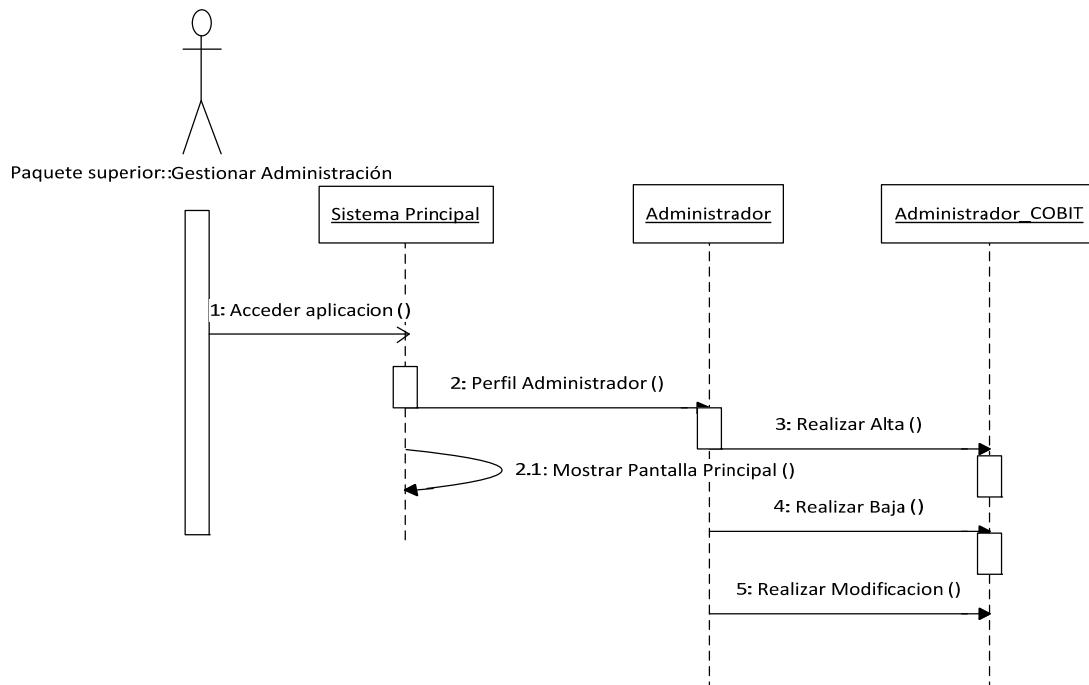


Ilustración 19. Diagrama de Secuencia Administración COBIT

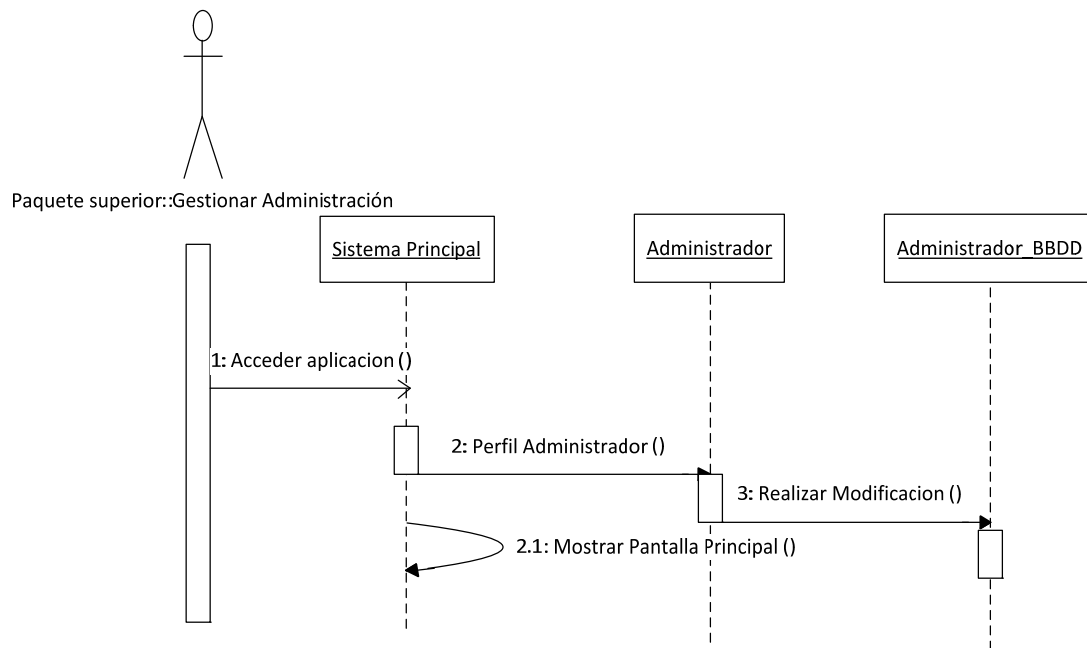


Ilustración 20. Diagrama de Secuencia Administración Base de Datos

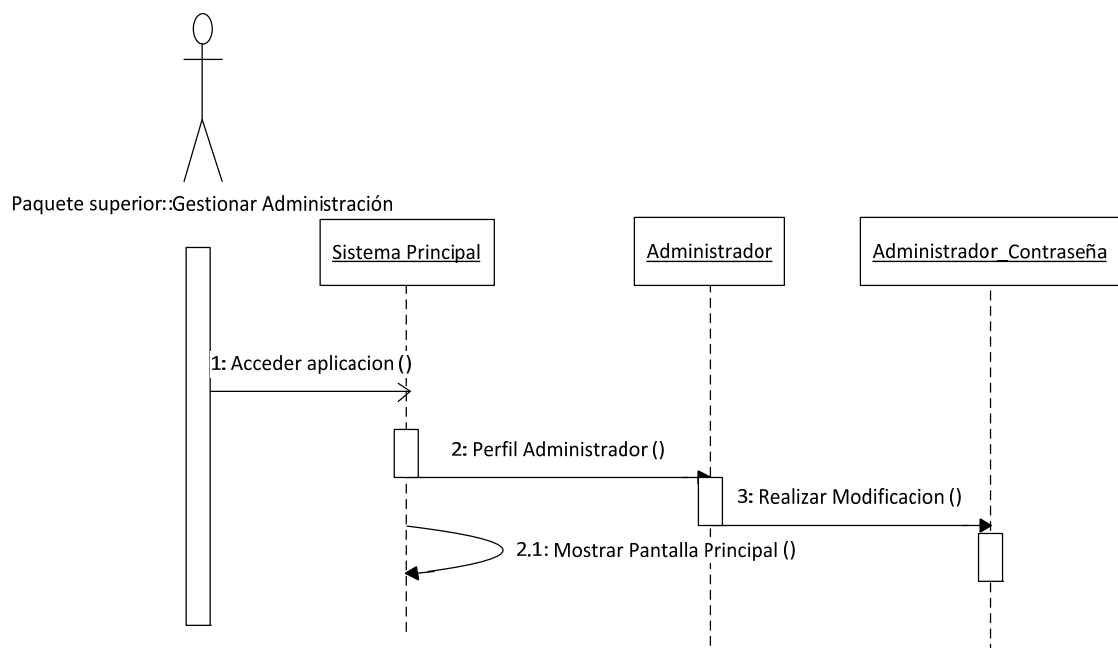


Ilustración 21. Diagrama de Secuencia Administración contraseña

5. Diseño Detallado

5.1 Diagrama de Clases UML

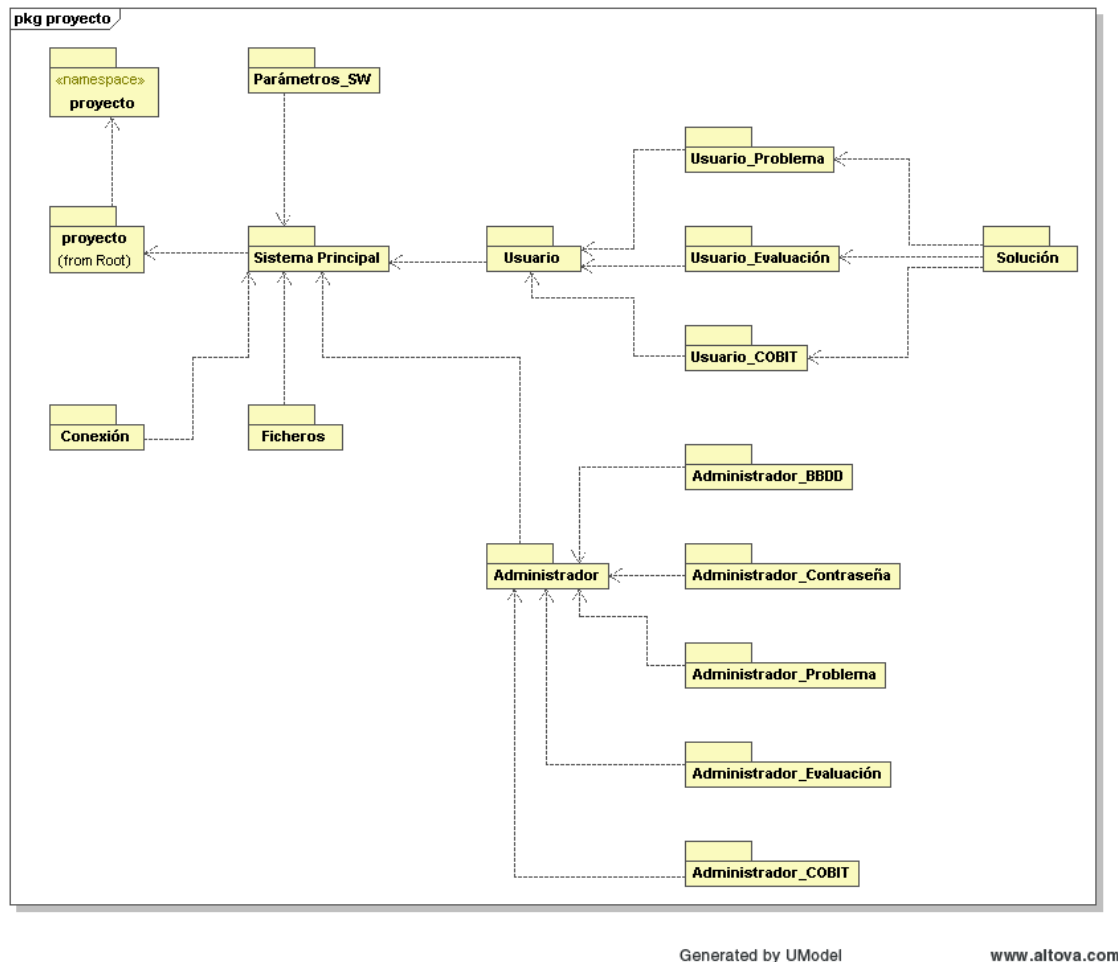


Ilustración 22. Diagrama de clases a nivel de paquetes

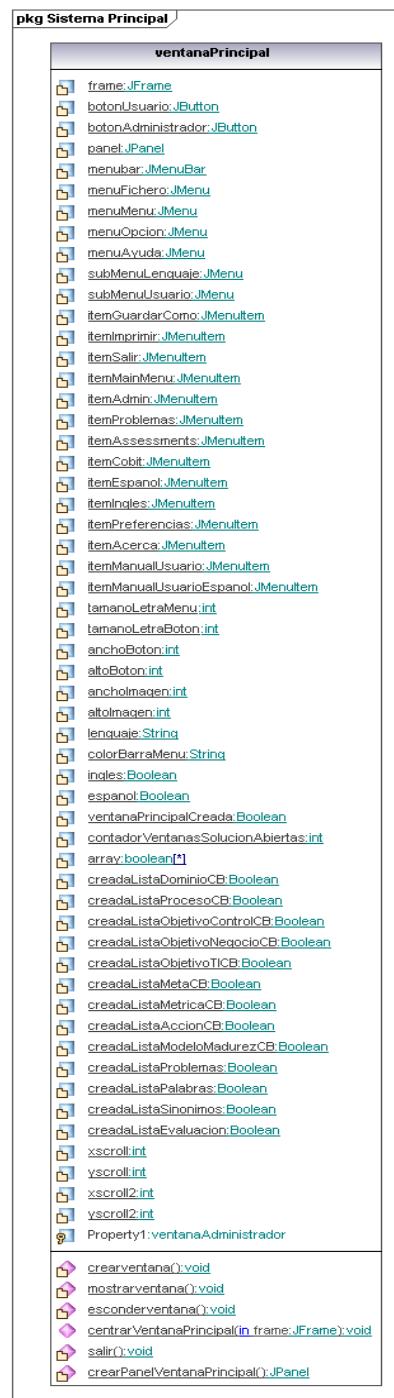
La figura anterior muestra el diagrama de clases a un nivel de paquetes. Cada paquete contiene una serie de clases que se detallarán en las siguientes páginas. El motivo de representar de esta forma el diagrama de clases es debido al tamaño del mismo, y además, a que de esta forma se obtiene una idea global mucho más clara que con un diagrama convencional de clases.

La relación entre los paquetes está mostrada por las líneas que los unen, pudiendo comunicarse entre ellos de forma que el programa construido funcione correctamente.

A continuación se detalla el contenido de las clases que componen cada paquete, y su función a nivel de funcionalidad.

5.1.1 Paquete Sistema Principal

Este paquete está compuesto por la clase **ventanaPrincipal**, que es la encargada de mostrar la pantalla principal del programa, a partir de la cual se puede ejecutar cualquier funcionalidad que es capaz de realizar este software.



Generated by UModel

www.altova.com

Ilustración 23. Paquete Sistema Principal

5.1.2 Paquete Ficheros

Este paquete está compuesto por las tres clases necesarias para el mantenimiento de los ficheros del programa. Se establecen operaciones de creación de los mismos, así como de modificación de su contenido. Es esencial para mantener los parámetros de configuración del software.



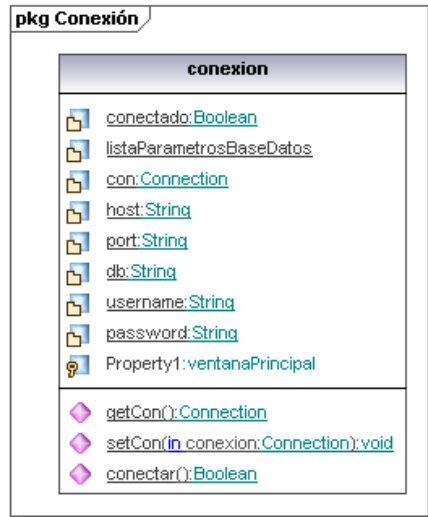
Generated by UModel

www.altova.com

Ilustración 24. Paquete Ficheros

5.1.3 Paquete Conexión

El siguiente paquete está compuesto por la clase **conexion**, que es la encargada de comunicar al programa con la Base de Datos de Oracle.



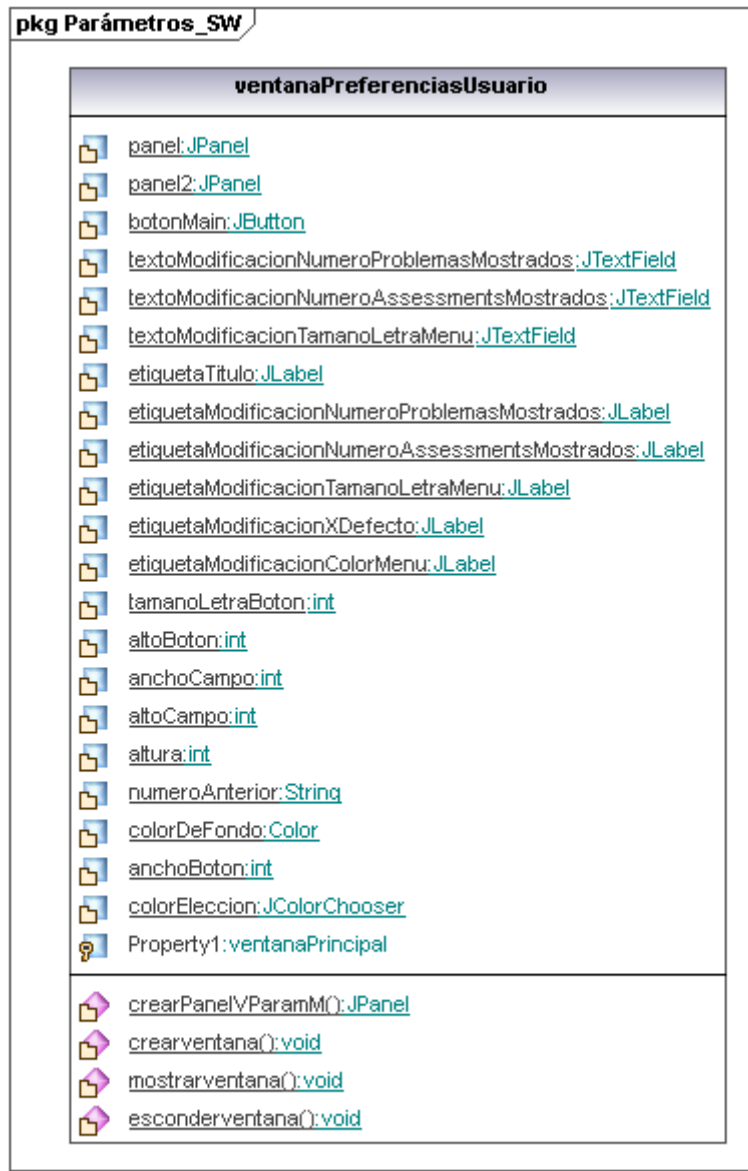
Generated by UModel

www.altova.com

Ilustración 25. Paquete Conexión

5.1.4 Paquete Parámetros_SW

Este paquete contiene la clase **ventanaPreferenciasUsuario**, una ventana desde la cual el usuario puede personalizar parámetros de la interfaz gráfica.



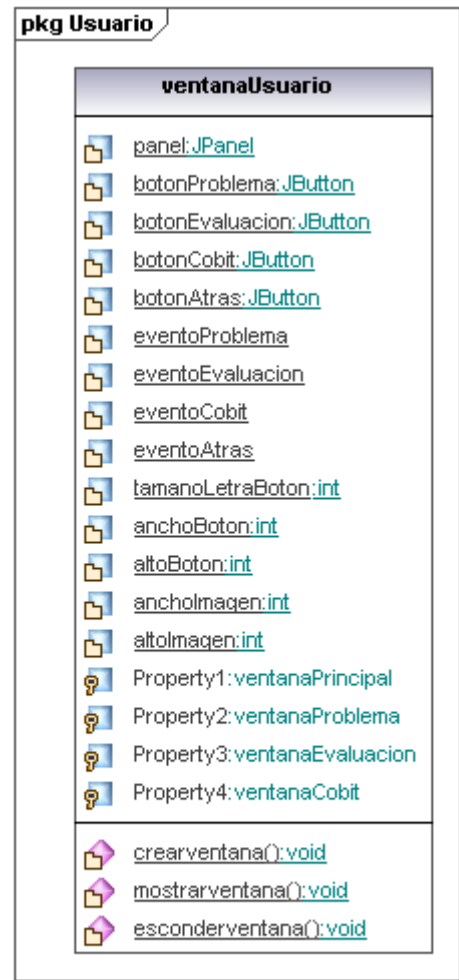
Generated by UModel

www.altova.com

Ilustración 26. Paquete Parámetros_SW

5.1.5 Paquete Usuario

El siguiente paquete contiene la clase **ventanaUsuario**, que es la encargada de mostrar por pantalla la ventana con el menú del usuario.



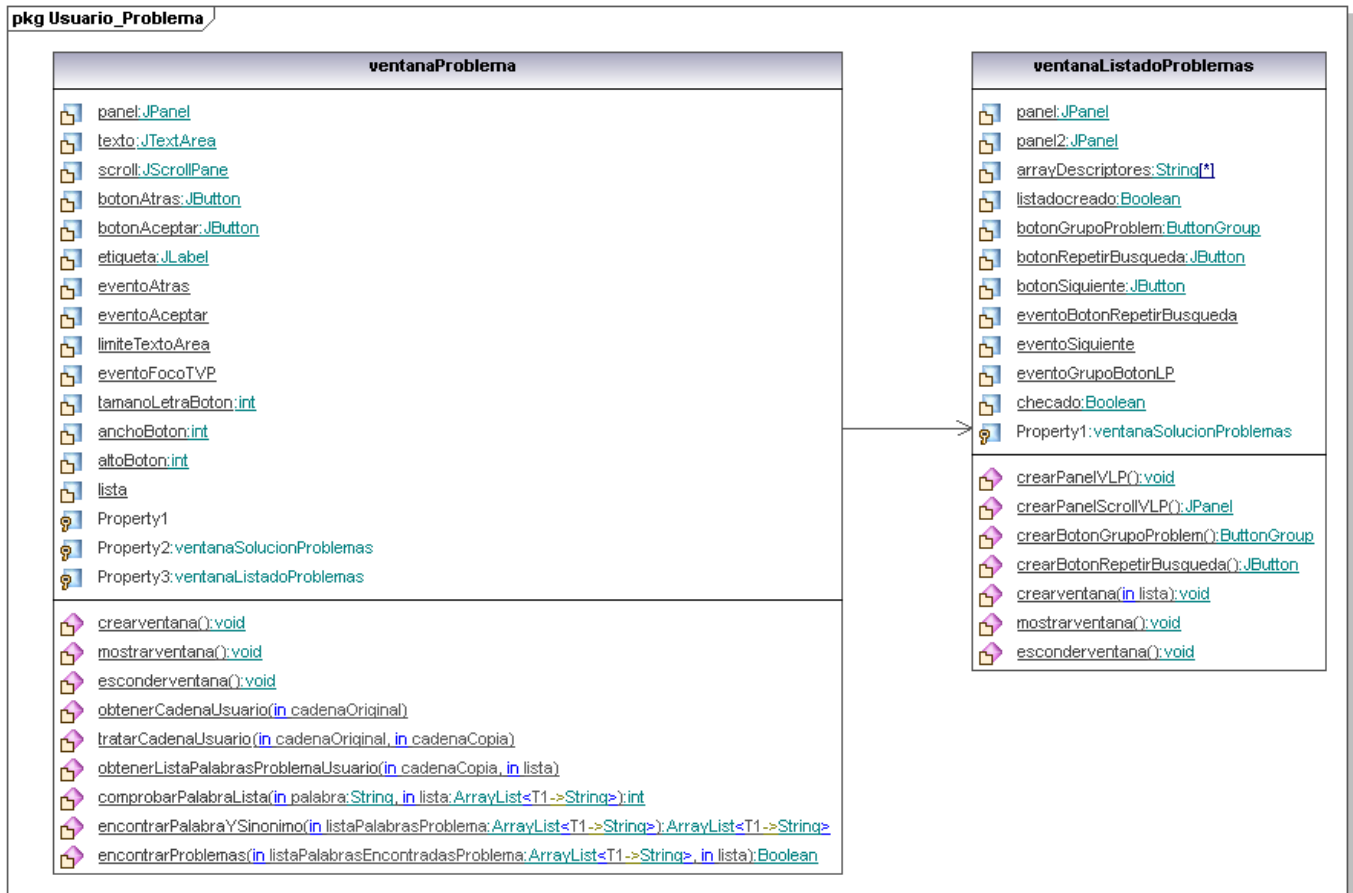
Generated by UModel

www.altova.com

Ilustración 27. Paquete Usuario

5.1.6 Paquete Usuario_Problema

El siguiente paquete está compuesto por las clases **ventanaProblema** y **ventanaListadoProblemas**. Mediante estas dos clases el usuario puede consultar los problemas encontrados a partir de uno que ha introducido previamente.



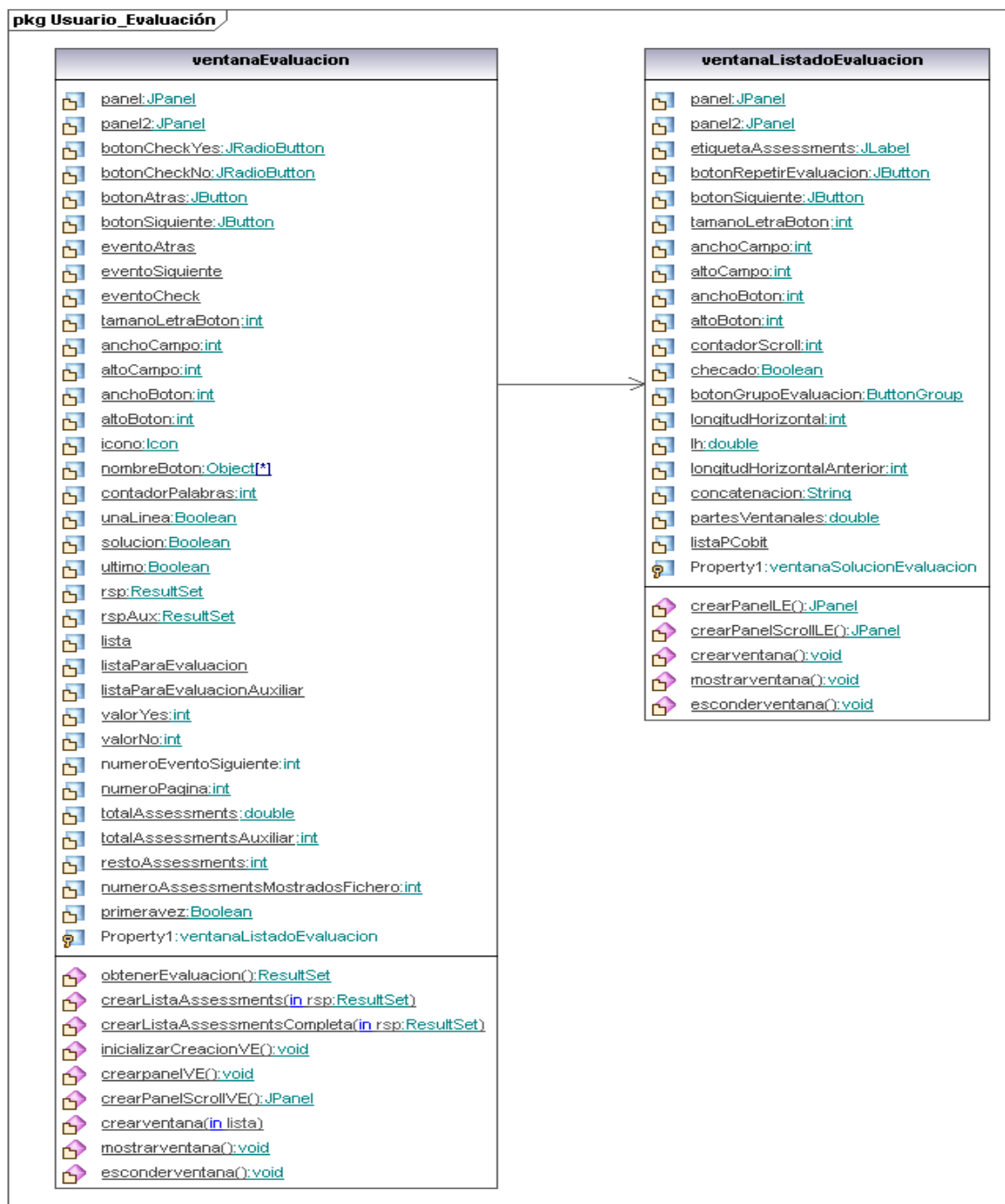
Generated by UModel

www.altova.com

Ilustración 28. Paquete Usuario_Problema

5.1.7 Paquete Usuario_Evaluación

Este paquete contiene las clases **ventanaEvaluacion** y **ventanaListadoEvaluacion**. A través de ambas el usuario puede realizar los tests de evaluación y obtener los resultados de los mismos.



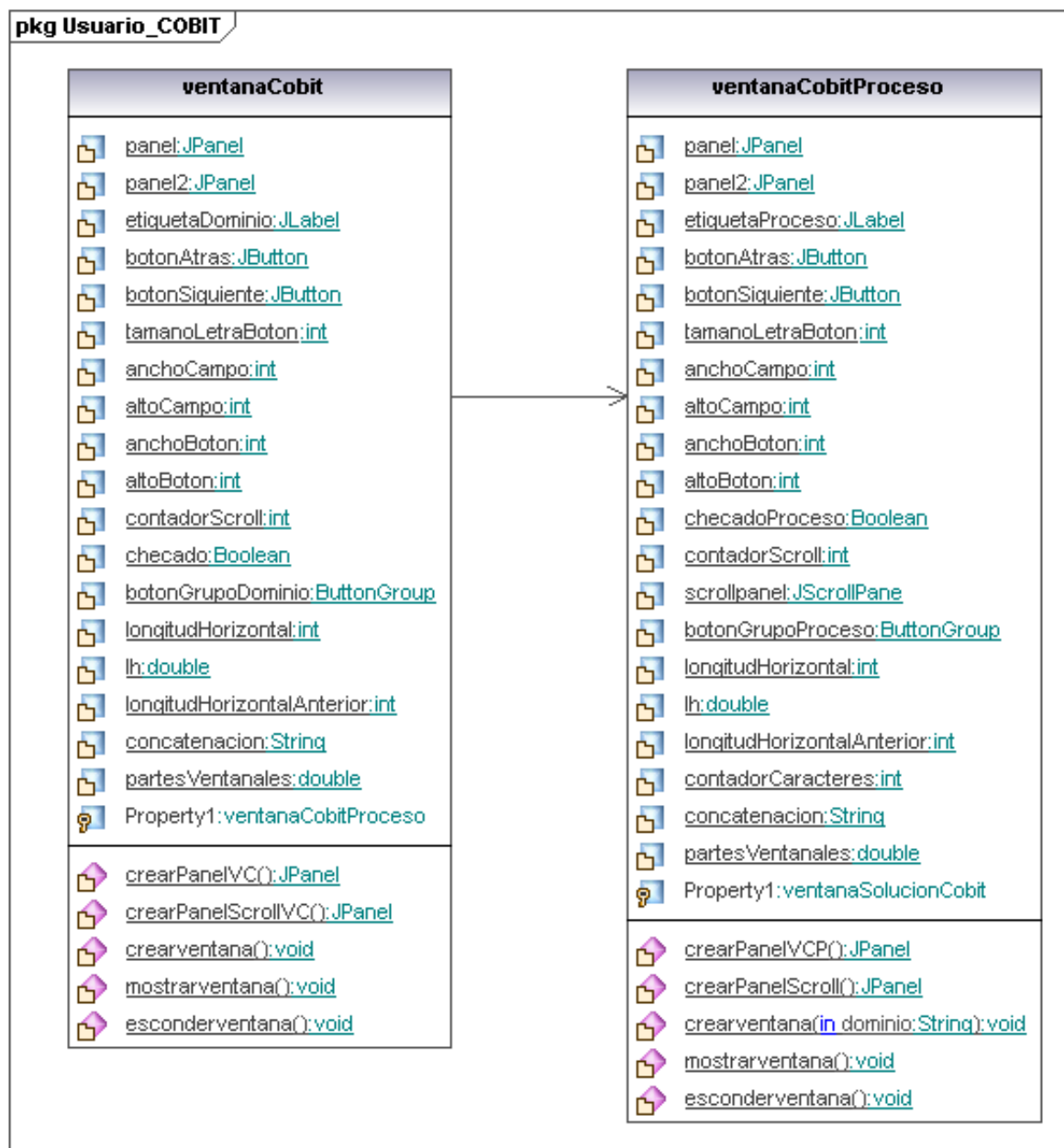
Generated by UModel

www.altova.com

Ilustración 29. Paquete Usuario_Evaluación

5.1.8 Paquete Usuario_COBIT

Este paquete contiene las clases **ventanaCobit** y **ventanaCobitProceso**. Mediante ambas clases el usuario puede consultar cualquier dominio y proceso de la metodología **COBIT 4.1**.



Generated by UModel

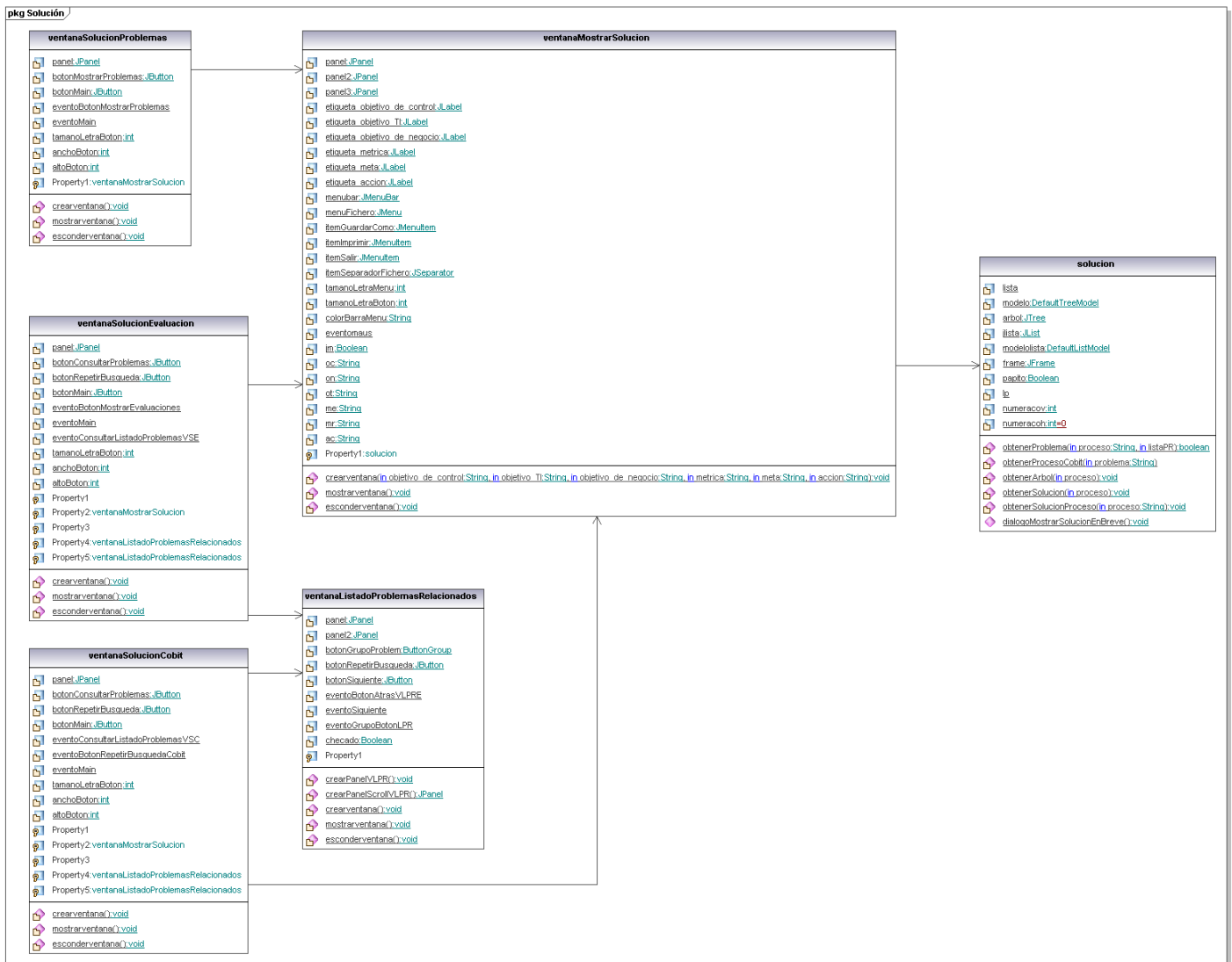
www.altova.com

Ilustración 30. Paquete Usuario_COBIT

5.1.9 Paquete Solución

El siguiente paquete contiene las clases necesarias para que se muestre la solución de las tres ramas posibles a las que tiene acceso el usuario para consultarla:

- Mediante la introducción de un problema.
- Mediante la realización de tests de evaluación.
- Mediante la consulta por dominio y proceso de **COBIT 4.1**.



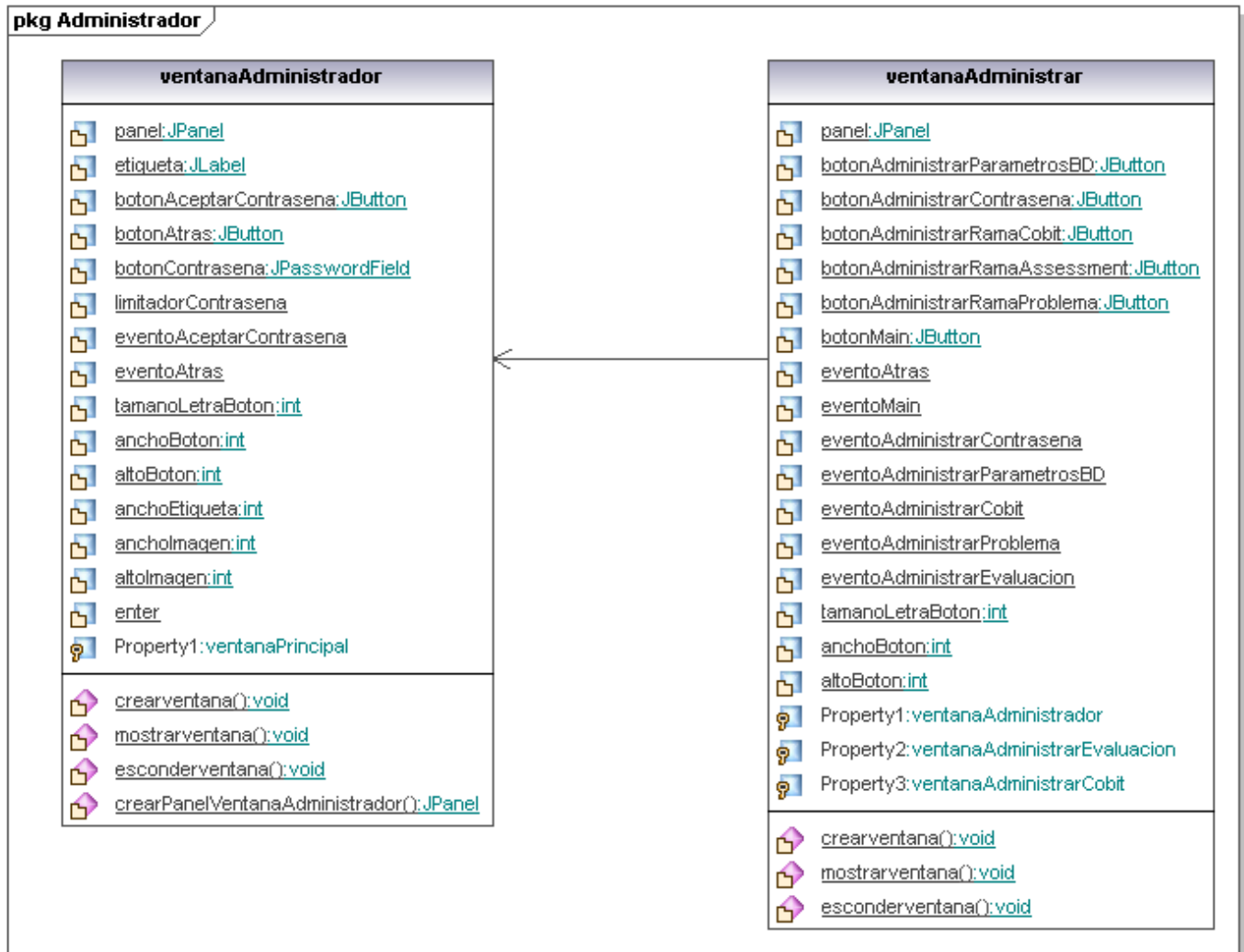
Generated by UModel

www.altova.com

Ilustración 31. Paquete Solución

5.1.10 Paquete Administrador

El siguiente paquete está compuesto por las clases **ventanaAdministrador** y **ventanaAdministrar**, que son las encargadas de mostrar el menú del administrador, desde el cual puede acceder a las funciones de administración del software.



Generated by UModel

www.altova.com

Ilustración 32. Paquete Administrador

5.1.11 Paquete Administrador_BBDD

El siguiente paquete está compuesto por las clases **ventanaConfigurarBaseDatos** y **ventanaConfigurarBaseDatosError**. Son las encargadas de establecer la configuración de la Base de Datos de Oracle con la que se conecta el programa.



Ilustración 33. Paquete Administrador_BBDD

5.1.12 Paquete Administrador_Contraseña

Este paquete contiene la clase **ventanaCambiarContraseñaAdministrador**, mediante la cual el administrador puede cambiar su contraseña de acceso a las funciones de administración.

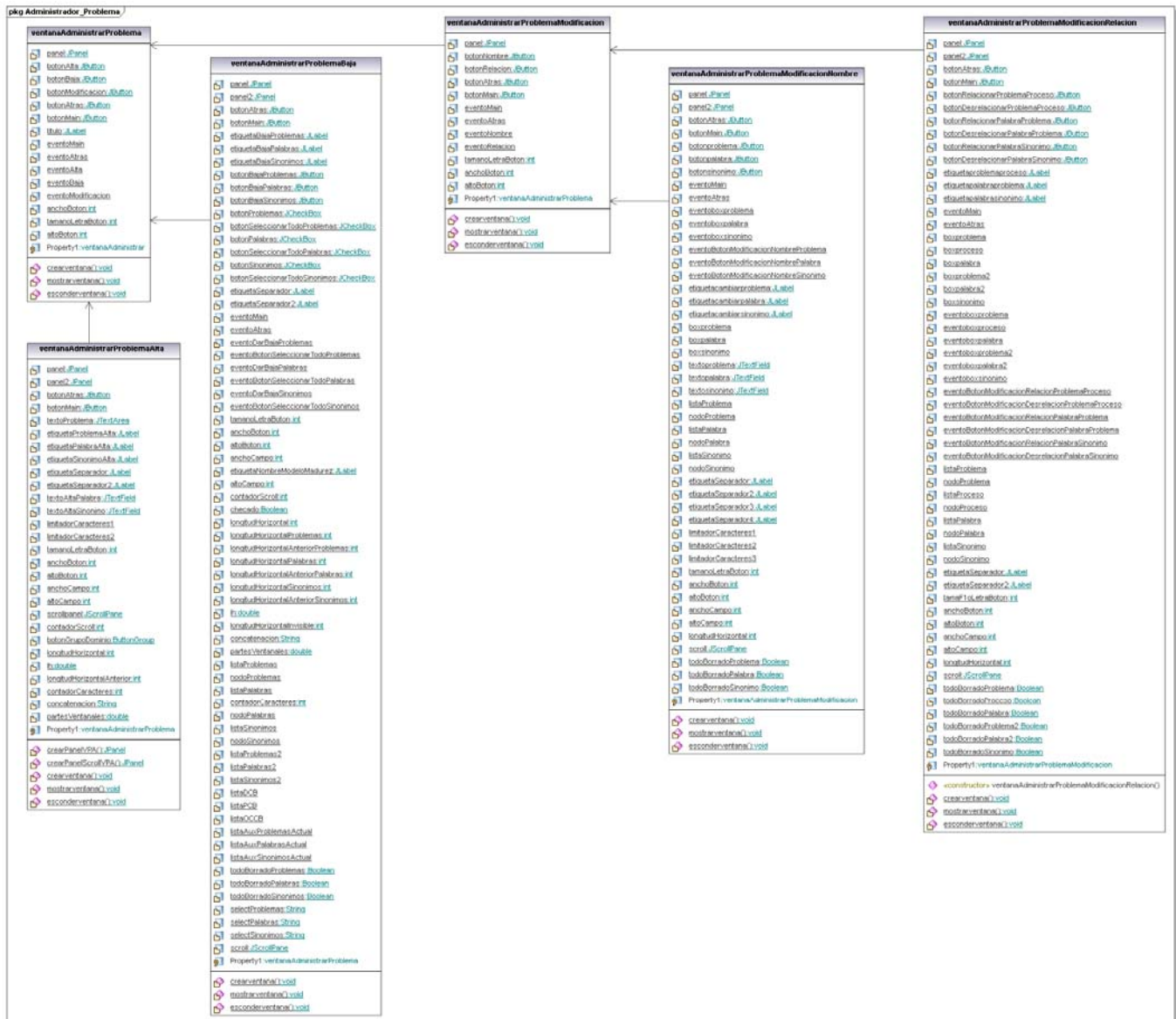


Generated by UModel

Ilustración 34. Paquete Administrador_Contraseña

5.1.13 Paquete Administrador_Problema

El siguiente paquete contiene las clases necesarias para que el administrador pueda administrar todos los parámetros relacionados con la rama de consulta de problemas del usuario.



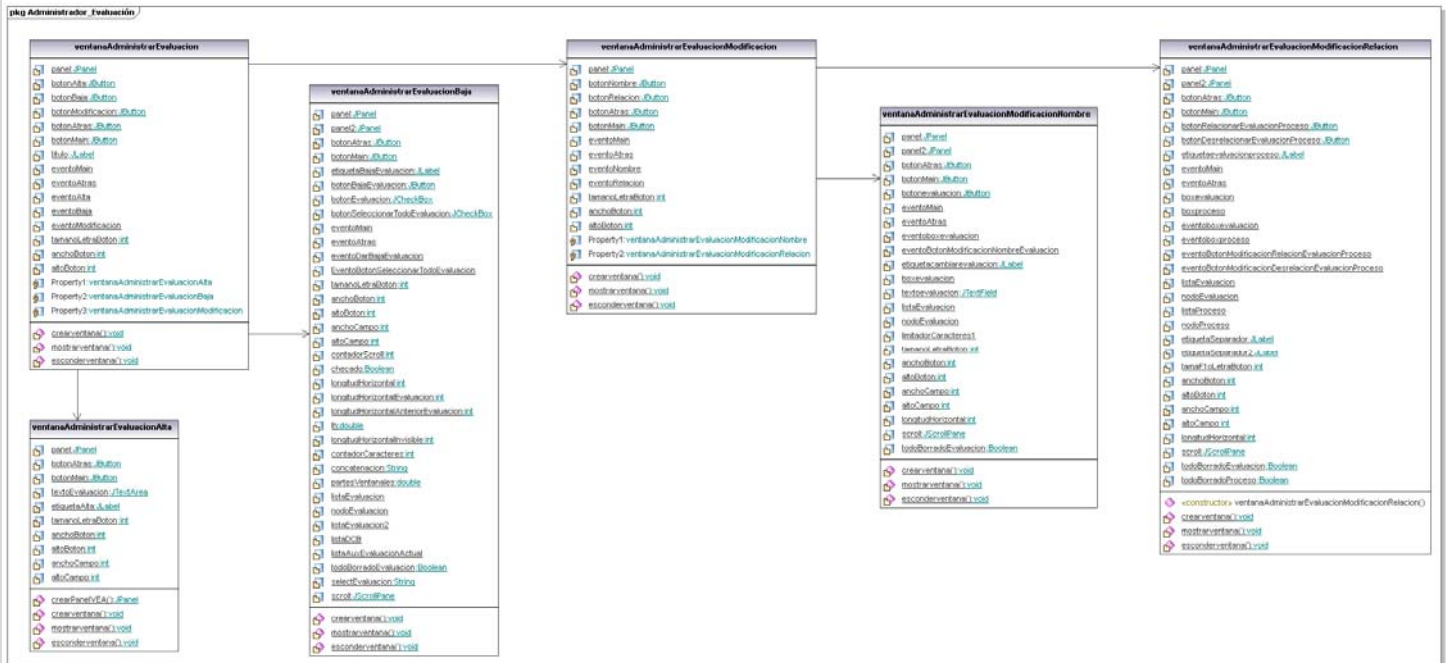
Generated by UModel

www.aitova.com

Ilustración 35. Paquete Administrador Problema

5.1.14 Paquete Administrador_Evaluación

El siguiente paquete contiene las clases necesarias para que el administrador pueda administrar todos los parámetros relacionados con la rama de tests de evaluación del usuario.



Generated by UModel

www.sitova.com

Ilustración 36. Paquete Administrador_Evaluación

5.1.15 Paquete Administrador_COBIT

Este paquete El siguiente paquete contiene las clases necesarias para que el administrador pueda administrar todos los parámetros relacionados con la rama de consulta del usuario por dominio y proceso de **COBIT 4.1**.



Ilustración 37. Paquete Administrador_COBIT

5.2 Esquema Entidad-Relación

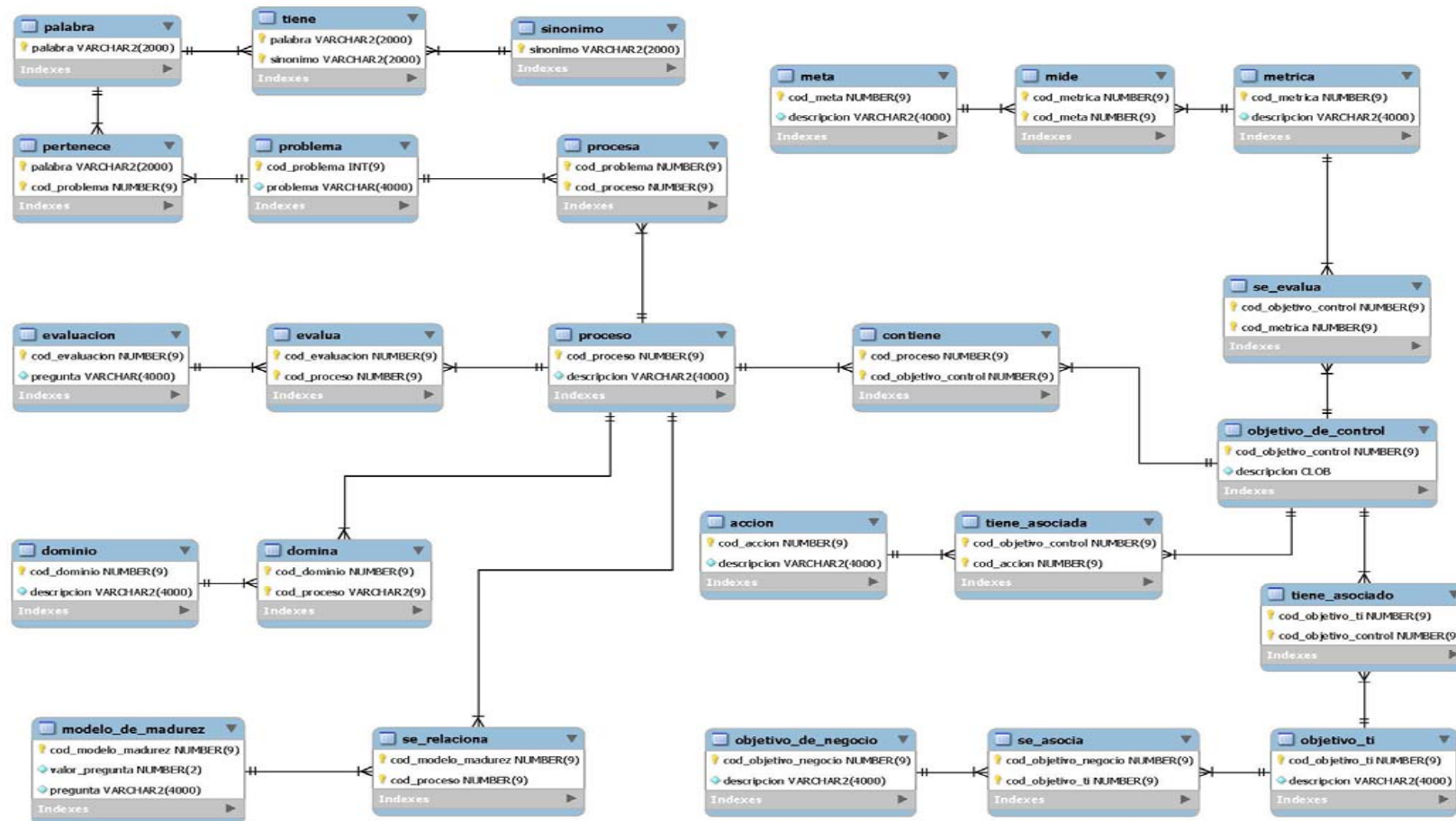


Ilustración 38. Esquema Entidad-Relación

5.3 Descripción detallada

La aplicación realizada en este proyecto se llama Information Technology Governance Solver (ITGS). Se ha elegido este nombre porque la aplicación es un *solucionador* de problemas de las Tecnologías de la Información dentro del ámbito del gobierno TI.

A continuación se detallan las características del diseño del aplicativo tales como el modelo de realización del ciclo de vida, el diseño de la interfaz gráfica, el funcionamiento interno de la aplicación, la forma en que se ha realizado el código fuente, y otros conceptos relacionados con las bases de datos como la inyección SQL.



Ilustración 39. Logotipo usado en el manual de usuario de ITGS. Registrado como Propiedad Intelectual. Copyright © 2012 Universidad Carlos III de Madrid. Todos los derechos reservados.

5.3.1 Ciclo de Vida

El ciclo de vida utilizado en el proyecto ha sido el modelo iterativo incremental, el cual está basado en varios modelos de cascada realimentados repetidamente de forma iterativa. Está fundamentado en la idea de realizar las diferentes tareas típicas del ciclo de vida como son el análisis, el diseño, la implementación, la integración y pruebas, y la operación y mantenimiento, en una serie de fases o iteraciones para ir depurando e incrementando el avance hasta llegar a la versión final.

Mediante este modelo se pueden ir mostrando poco a poco los avances en la aplicación y a su vez poder hacer uso de reutilización y tener la posibilidad de realizar cambios sin mucho coste, excepto si el proyecto está muy avanzado, por la funcionalidad que da el ir incrementando el proyecto fase a fase.

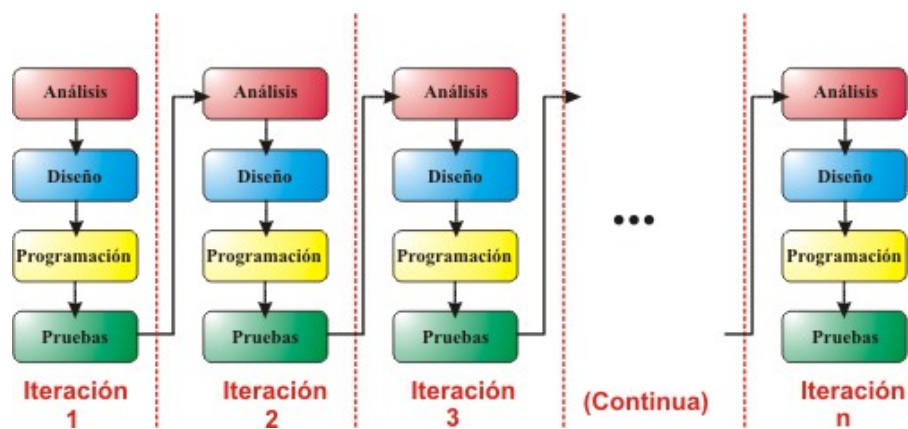


Ilustración 40. Diagrama del modelo iterativo incremental

La entrega del software bajo este modelo se realiza en partes funcionales y reutilizables más pequeñas que se llaman incrementos. Tras cada incremento se construye el siguiente, generalmente sobre el anterior, y así en cada integración se obtiene un producto con mayor funcionalidad que el anterior. Este proceso es repetido hasta llegar al software final completo.

Este enfoque es muy útil cuando se dispone de poco personal para el desarrollo o si no hay fecha límite de entrega y también a fines de versiones de evaluación ya que las entregas incompletas proporcionan en un principio al usuario una funcionalidad básica y cada vez mayor.

Este modelo permite probables y esperados cambios en los requisitos en tiempo de desarrollo, es decir que admite cierto margen para que el software pueda evolucionar durante el desarrollo. Esto es útil especialmente cuando los requisitos son medianamente bien conocidos pero no son completamente estáticos y definidos, en cuyo caso sería mejor usar un modelo en cascada. También es aconsejable para

desarrollos en los que se observe que posee áreas bastante bien definidas con suficiente independencia entre ellas con un posible tratamiento de prioridad.

El modelo permite por tanto una implementación con refinamientos sucesivos de ampliación o mejora, de tal manera que con cada incremento se agrega nueva funcionalidad, se cubren nuevos requisitos o se mejora la versión previamente implementada. De este modo se brinda cierta flexibilidad para que durante un desarrollo se incluyan cambios en los requisitos por parte del usuario, pero si el software se encuentra en un estado muy avanzado podría afectar fuertemente en los costos por lo que es importante realizar un acuerdo con el cliente. Otro aspecto positivo de este modelo es que permite realizar entregas funcionales tempranas al cliente.

El modelo no obliga a especificar con precisión y detalle absolutamente todo lo que el sistema debe hacer. Esto hace más manejable el proceso reduciendo el impacto en los costos y en caso de alterarse los requisitos solo afecta a una parte del sistema, aunque esto se puede agravar si los cambios son en los últimos incrementos, en un estado avanzado. Se basa en un desarrollo modular con entregas parciales que son escogidos según prioridades definidas de alguna manera, ya sea por razones del usuario o del desarrollador del software.

Este modelo no es recomendable para sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido o de alto índice de riesgos.

En el caso del proyecto es el modelo que mejor se adapta ya que los requisitos no fueron definidos en detalle, además de no ser bien conocidos desde el principio, sino que cambiaron y fueron siendo más específicos a lo largo del tiempo, permitió por tanto flexibilidad y realizar entregas funcionales parciales para poder ir viendo el desarrollo poco a poco. La estructura del sistema era modular ya que sí existía una división de los diferentes componentes del sistema como la gestión de los usuarios, usuario y administrador, consultas por tres ramas distintas... Además permite que en el futuro se puedan seguir realizando cambios en el sistema.

5.3.2 Interfaz gráfica de usuario

¿Por qué la manera en que se ven las cosas es tan importante?

El diseño de la ventana de la aplicación y su tamaño, el texto, el color, las imágenes, las imágenes en movimiento o videos, los sonidos... interactúan para producir un ambiente agradable al usuario. Por tanto es importante hacer un uso adecuado de cada uno de estos componentes ya que tienen un impacto significativo hacia el usuario, e influenciará en su opinión sobre un producto.

El diseño de la interfaz es crítico para el manejo de una aplicación ya que si el diseño es bueno, es decir que incorpora controles intuitivos y de fácil manejo, el usuario podría utilizar correctamente el programa sin la necesidad estudiar un manual o recibir formación de un experto.

Para el diseño de la interfaz gráfica, interfaz de usuario (UI, User Interface) o interfaz gráfica de usuario (GUI, Graphic User Interface) de la aplicación se han tenido en cuenta estas características para proveer de una interfaz agradable, sencilla, fácil, cómoda y en definitiva que sea usable y eficiente para el usuario, a la vez que se han aplicado ciertos modelos o guías para el tratamiento de los distintos componentes. La interfaz de la aplicación está basada en la filosofía de diseño conocida como Diseño centrado en el usuario o User-centered Design (UCD) en la que el usuario es considerado como el centro del diseño. Los usuarios y diseñadores trabajan juntos para conseguir el producto deseado teniendo en cuenta limitaciones posibles. Es un diseño participativo en el que el usuario contribuye en el diseño del sistema. Entre las principales áreas que se tienen en cuenta para el diseño se encuentran la visibilidad (legibilidad, estructura), accesibilidad (legibilidad, navegación, simplicidad), interacción (consistencia, objetos)... Aparte de esto las guías de diseño de interfaces de usuario usadas han sido:

- Designing the user interface: text, colour, images, moving images and sound de OpenLearn LabSpace que está basado en guías de Götz (1998), Hartley (1994) and Rivlin et al. (1990) que tratan sobre el diseño de texto, colores y pantallas.
- GUI Look and Feel Standard desarrollado por DENR ITS. Está basado en Microsoft's [Web site](http://www.microsoft.com/win32dev/uiguide/default.htm) (<http://www.microsoft.com/win32dev/uiguide/default.htm>); "The Windows Interface Guidelines for Software Design" published by Microsoft Press; "GUI Design Essentials" by Weinschenk, Jamar and Yeo.
- The Windows Interface Guidelines — A Guide for Designing Software. Microsoft Windows. February 1995.

Es importante tener en cuenta el tipo de aplicación que se va a desarrollar para pensar en un tipo de interfaz u otra. Por ejemplo para la mayoría de aplicaciones se

asume que se quiere comunicar la información de una manera simple pero eficiente. En cambio si la aplicación es una web de diseño gráfico quizá la interfaz tenga que ser innovadora.

La aplicación del proyecto es una aplicación totalmente informativa, ya que su principal objetivo es mostrar información que ayude a las empresas a solucionar problemas TI. Por tanto se necesita una interfaz que comunique de la mejor manera posible la información, es decir que lo lógico sería el uso de una interfaz simple sin muchos adornos que despisten de la función principal que es la de mostrar información de una manera eficiente al usuario.

En cuanto al tamaño de la ventana (marco, en Java JFrame) de la aplicación, se debe pensar siempre en el hardware sobre el que se va a implantar o sobre el que podría ser implantado. Hoy en día con el hardware actual, el tamaño debería ser como mínimo de 800 por 600 píxeles (800x600). En la aplicación del proyecto se ha usado un tamaño de 700x700 píxeles por dos motivos. El primero es que este tamaño es visible en monitores que sean pequeños y grandes, en la mayoría de monitores, a no ser que se usara un monitor de menos de 10 pulgadas aproximadamente, la ventana se vería completamente en la pantalla del monitor. El segundo motivo es que en un principio esta aplicación iba a ser integrada en una página web y uno de los requisitos es que no ocupara una gran porción de la pantalla, por lo que un tamaño cuadrado de 700x700 era idóneo para representar la información sin dificultad de lectura y sin ser excesivamente grande ni pequeño el tamaño de la ventana. También por este motivo de la integración en página web, la aplicación no permite maximizar la ventana, además si se permitiera esto, la distribución de botones, etiquetas... se vería afectada y habría que reorganizarla de otra manera. Únicamente se permite minimizar la ventana y usar el icono X para salir de la aplicación teniendo el botón de maximizar inhabilitado. Sin embargo, cuando usamos la aplicación y obtenemos una solución, en ese momento se abre otra ventana y esta sí se puede maximizar ya que tiene una gran cantidad de información y para leerla correctamente se permite maximizar la ventana al tamaño del monitor que se esté usando. Además de poder maximizar también se puede minimizar y salir con el icono X.

Esta ventana es el marco de la aplicación, es el límite de la aplicación, el JFrame. Asociado al JFrame se encuentra el JMenuBar que es la barra de menú o barra de herramientas en Java. Y dentro del JFrame, incluido en él, el resto de porción, donde se puede encontrar cualquier componente de Java ya sean botones, cajas de texto, barras de desplazamiento..., se llama panel, en Java JPanel, el cual lleva incluidos los componentes citados. A partir de ahora, cuando se hable de ventanas se estará haciendo referencia indistintamente al JFrame o al JPanel. Si nos ajustamos a la definición cuando se habla de ventanas se estaría incluyendo todo, tanto los límites de la aplicación, como la barra de menú y el propio panel, y si se hablara de paneles se estaría hablando de la ventana pero sin incluir la barra de menú, ni la barra de título. Pero la nomenclatura que se seguirá es la de usar ventana en ambos casos ya que es más descriptiva y sencilla y al fin y al cabo los paneles forman parte de la ventana con

la mayor porción de ella. Además las clases en el código se nombraron como ventanas y no como paneles por el mismo motivo. Pero internamente, la aplicación solo consta de una única ventana el JFrame principal (la ventana solución también tiene su propio JFrame ya que es una ventana independiente), lo que cambian son los paneles (JPanel) asociados al JFrame que van siendo actualizados de acuerdo a los eventos que el usuario va produciendo sobre los distintos paneles asociados todos ellos a un único JFrame. Pero como hemos dicho a partir de ahora ventana no se referirá al único JFrame principal sino a la ventana con su panel en un momento determinado de la aplicación. Esta es la razón de que la barra de menú aparezca en todas las situaciones ya que pertenece al JFrame principal y no al JPanel que sí va cambiando.

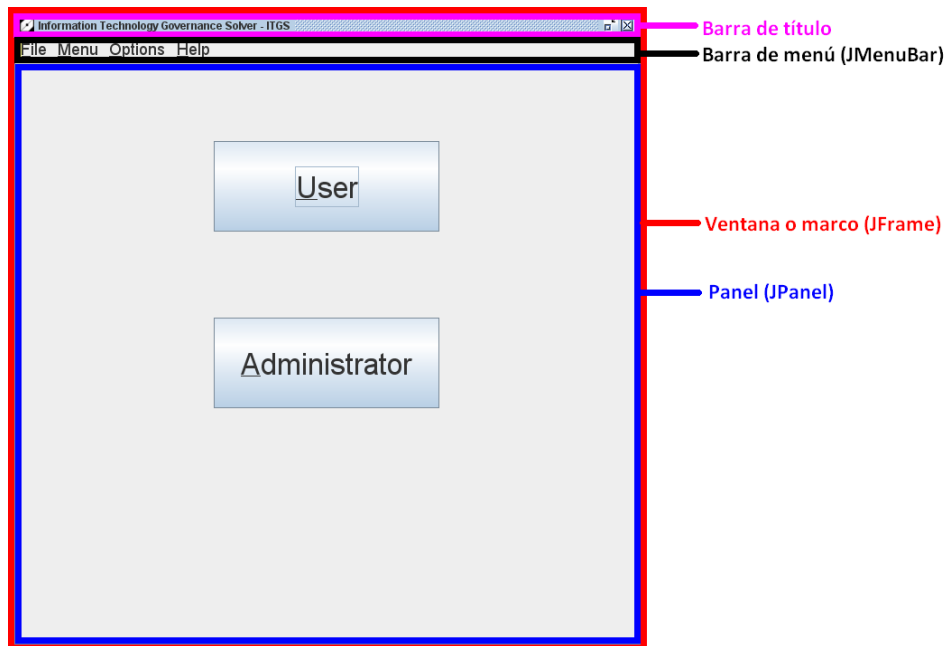


Ilustración 41. Estructura de la ventana de la aplicación

Para la convención de nombres se ha usado en cada ventana el nombre de la aplicación: Information Technology Governance Solver – ITGS seguido de una barra de menú que aparece en todas las ventanas de la aplicación y consta de File (Archivo), Menu (Menú), Options (Opciones) y Help (Ayuda).

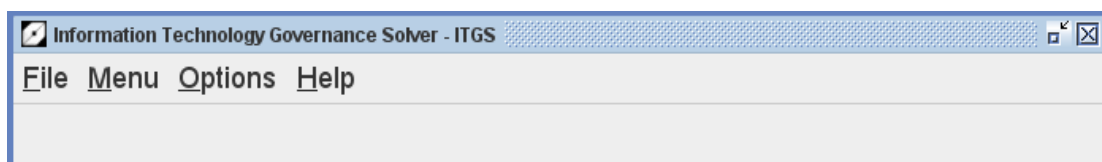


Ilustración 42. Encabezado de la ventana aplicación con su menú y título

La barra de menú en la ventana de la solución cambia y solo tiene el menú File (Archivo) puesto que las opciones en este caso únicamente son la de guardar la

solución en fichero, imprimir y salir. Además permite maximizar la ventana al contrario que en las otras ventanas de la aplicación.

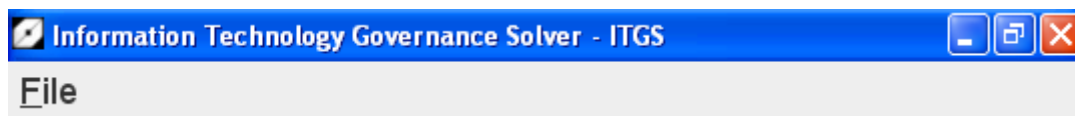


Ilustración 43. Encabezado de la ventana aplicación con su menú y título en la solución

El texto en una aplicación es una de las más importantes y poderosas componentes ya que es el principal modo de comunicación de información. Para el texto usado, deberíamos asegurarnos de que, tanto la fuente como el tamaño del texto fuera legible y adecuado. En la aplicación se ha usado una de las fuentes que posee Java llamada fuente “Dialog”. Entre otras fuentes de Java están “Serif” o “Sans-Serif” por ejemplo. El tipo de letra “Dialog”, al igual que “Sans-serif”, es una fuente legible y puede usarse para un buen entendimiento del texto que se muestra en la aplicación.

El tamaño de la letra debería estar en el rango de 8 y 72 puntos, que es un rango bastante grande. Por lo general el tamaño de la letra en la aplicación varía muy poco siendo casi el mismo en toda la aplicación, sin ser excesivamente pequeño ni excesivamente grande, variando en un rango de entre 18 a 33 puntos, de esta forma se logra evitar la confusión con muchos tamaños distintos de letra y concentrar la atención del usuario. Una de las opciones del programa permite aumentar el tamaño de la letra del menú, esto podría facilitar su uso a personas con problemas de visión. El rango disponible en el tamaño de la letra de menú va desde 10 a 40 puntos. No se permite introducir un valor menor de 10 puntos ya que la visibilidad sería muy limitada y tampoco más de 40 puntos ya que excedería en demasía el tamaño. En el capítulo de “Líneas futuras” se explican más opciones de accesibilidad para gente con ciertas discapacidades.

Se ha evitado hacer uso de la letra cursiva y del subrayado ya que pueden dificultar la lectura de texto.

El color usado para la letra es el negro sobre fondo blanco o gris. Si se usa una fuente con otro tipo de color se recomienda realzarlo en negrita porque es más difícil de leer que texto en color negro.

Para enfatizar ciertos campos se ha usado negrita, únicamente en lugares para facilitar la comprensión y la lectura. En general no se recomienda enfatizar por color sino por negrita, porque usualmente los usuarios pueden confundir los colores con otras características especiales como el color utilizado en los hipervínculos.

También se ha usado un interlineado algo más amplio de lo normal para facilitar la lectura, especialmente en el caso de la solución, y un ancho de línea adaptado al ancho de la pantalla en la mayoría de los casos, si bien en caso de no ajustarse a la pantalla se tiene la opción de utilizar una barra de desplazamiento (scroll, en Java JScrollPane) para poder leer la información entera, tanto vertical como horizontalmente. Lo mismo ocurre cuando guardamos la solución en un archivo externo, tiene un interlineado y un ancho de línea que ocupa la pantalla para facilitar

su lectura. En algunos casos de administración las líneas no tienen salto de línea para no entorpecer el resto de la información administrativa.

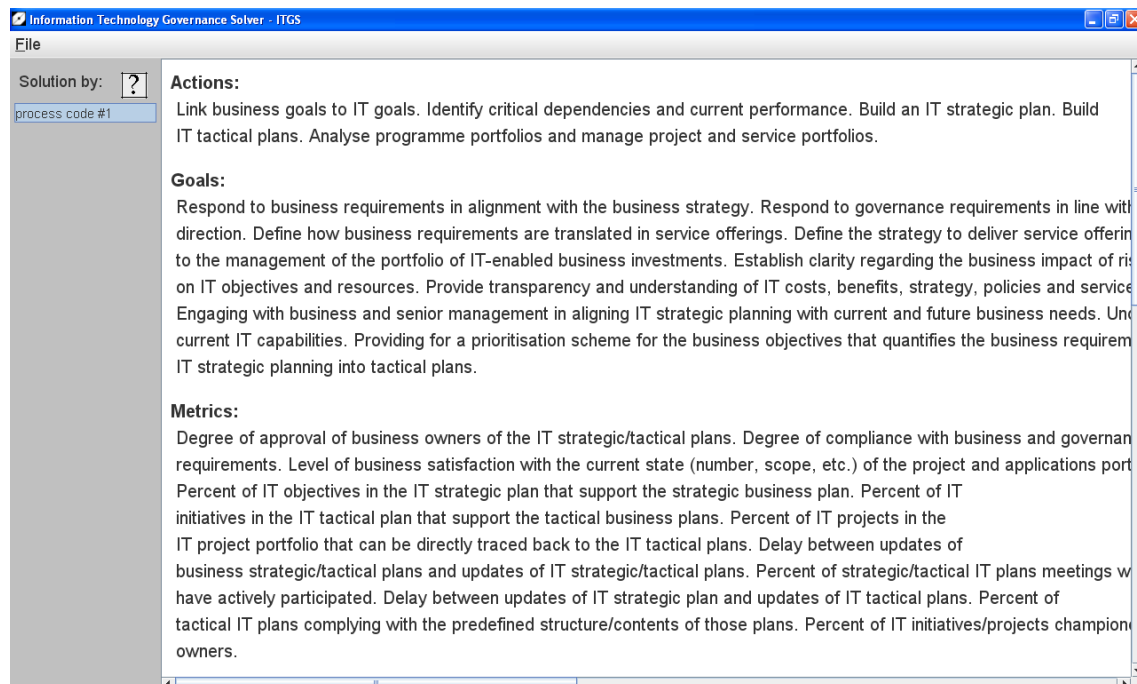


Ilustración 44. Solución de ITGS con tipo de fuente Dialog, fondo blanco y barras de desplazamiento vertical y horizontal (scroll)

El color es un componente importante para captar la atención del usuario. Se pueden usar ciertos colores para transmitir con mayor efectividad cierta comunicación pero debemos asegurarnos de que los colores usados tengan unas connotaciones correctas.

Si se usan otros colores distintos a los típicamente usados como el blanco, negro o gris hay que tener en cuenta que su significado puede estar ligado a ciertas connotaciones culturales y que pueden no ser las mismas en distintos países. Por ejemplo el color rojo es típicamente entendido como un color que representa peligro en el mundo occidental pero que en China es entendido como un color alegre.

Colores	Connotaciones positivas	Connotaciones negativas
Rojo	Activo, estimulante, emocionante, poderoso, fuerte, enérgico, atractivo, dominador	Agresivo
Azul	Controlado, misterioso, intelectual, armónico, profundo, soñador, fiel, racional, sensato	Agresivo, distante, melancólico
Azul/Verde/Turquesa	Reconfortante	Distante, obstinado, carente de emociones, frío, estéril
Verde	Reconfortante, armonioso, optimista, cercano a la naturaleza, tranquilo, amable, conciliador, de carácter fuerte	Celoso, envidioso, inexperimentado
Amarillo	Colorido, extrovertido, alegre, jovial, divertido, luminoso	Superficial, exagerado, vano
Naranja	Emocionante, directo, alegre, vivo, comunicativo, cálido	Íntimo, enérgico, posesivo, barato
Rojo oscuro	Elegante, serio, digno	Arrogante

Tabla 70. Connotaciones de los colores (Götz (1998) resumido por OpenLearn LabSpace)

El color de fondo mayoritario de la aplicación es el color gris. Este es el color que por defecto Java aplica a sus ventanas. Se trata de un color que facilita la lectura con texto negro y es el estándar utilizado en la mayoría de aplicaciones. Para los “box” y otras casillas o cajas de texto en las que se permite introducir datos por parte del usuario, el color que por defecto aplica Java es el blanco. Este color facilita al usuario el entendimiento de que puede introducir texto y es el color estándar para la mayoría de aplicaciones. Para la ventana de solución se ha utilizado el color blanco y no el gris, ya que facilita su lectura con texto en negro puesto que el texto es mucho mayor que en las otras partes del programa, y también gracias a un mayor contraste entre texto y fondo, e imita de cierta manera a un visor típico de texto como los visores de PDF, Word... cuyos fondos son típicamente blancos ya que facilitan la lectura. En las opciones de preferencias del programa se permite cambiar el color de la barra del menú para personalización del usuario, o para facilitar lectura a usuarios con problemas de visión. El usuario puede elegir entre una amplia paleta de colores y en cualquier caso el usuario puede volver a la versión por defecto rápidamente por si el efecto causado no es el idóneo.

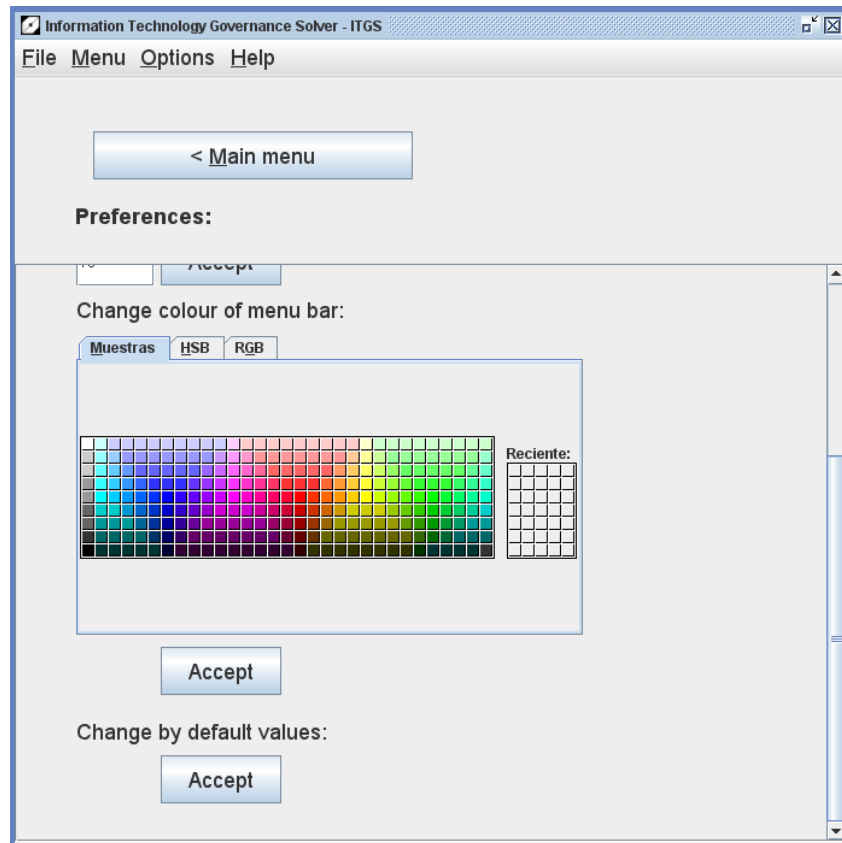


Ilustración 45. Paleta de colores para la personalización de la barra de menú y botón para cambio de valores por defecto

Si se usan colores para delimitar o agrupar ciertas partes de la ventana, hay guías de diseño de interfaces que recomiendan no usar más de seis colores distintos en la pantalla ya que podrían crear confusión y ser desagradables a la vista. Algunos estudios afirman que el uso arbitrario de muchos colores puede provocar vista cansada, fatiga y desorientación en los usuarios. En la aplicación no ha sido necesario utilizar un gran número de colores para identificar correctamente las distintas zonas. En general se usan fondos grises y blancos y los botones, al igual que las barras de desplazamiento, tienen el color que por defecto Java aplica que es un azul claro casi cercano a un gris con degradado que le da un efecto de brillo como si fuera un objeto tridimensional para facilitar la comprensión al usuario de que es un objeto en el que se puede hacer *click*. Los bordes de la ventana de la aplicación son de color azul más oscuro y son uno de los que Java pone por defecto para delimitar el área de la aplicación.

Otro factor importante a la hora de usar el color es el brillo y saturación del color, es decir el contraste de unos colores con otros. Por ejemplo si usamos un fondo blanco y sobre él usamos texto de color amarillo va a resultar muy difícil su lectura, en cambio si sobre un fondo blanco aplicamos texto negro la lectura va a ser ideal por el contraste creado. Por ello los colores empleados en la aplicación son texto negro sobre

fondos grises y blancos, que crean, como hemos dicho con anterioridad, mayor contraste facilitando la visión de la información.

Porcentaje	Colores acromáticos	Colores	Luminosidad
100	Blanco		Muy alta
		Amarillo, amarillo-verde	Alta
		Naranja	Alta a media
	Gris medio	Rojo, verde	Media
		Violeta	Media a baja
		Azul	Baja
0	Negro		Ninguna

Tabla 71. Luminosidad de los colores. (Götz (1998) en OpenLearn LabSpace)

Además de permitir el cambio de color de la barra de menú y el tamaño de la letra, en las preferencias de la aplicación se permiten otras opciones de personalización para el usuario.

Por ejemplo es posible cambiar el lenguaje de inglés a español y viceversa, aunque como se explica en el apartado “Líneas futuras” esta opción solo está disponible a modo de prueba en unas pocas ventanas. También es posible configurar el número de problemas y problemas relacionados en la solución que el usuario desee que se muestren por pantalla. Es decir, puede que el número de problemas encontrados sea de 20 pero el usuario solo quiera ver 5, o todos los posibles, entonces en preferencias el usuario puede configurar el número deseado. Si pone un número mayor de problemas que quiere visualizar del número máximo de problemas que se encuentran en la base de datos, la aplicación le avisará y colocará el valor máximo de problemas que existen en la base de datos. A su vez el valor mínimo permitido es ‘1’ ya que no tiene sentido que se muestren cero problemas.

Por otro lado se permite cambiar el número de evaluaciones que se muestran por pantalla cuando realizamos la consulta por la rama de usuario de los assessments para conocer la madurez de la empresa. Al igual que ocurre en la opción anterior, el valor ‘0’ no está permitido ya que carece de sentido realizar cero evaluaciones por ventana y el valor máximo está igualmente controlado por el valor máximo que se tiene en la base de datos del número de evaluaciones existentes. Cuando el usuario elija un número determinado y correcto, la ventana de la rama de las evaluaciones se adaptará para mostrar el número correcto de evaluaciones por ventana, si el usuario elige el mayor número posible para mostrar, la ventana se adaptará para mostrar todas las evaluaciones en una única ventana ya que no será necesario dividir las evaluaciones en varias ventanas.

También se permite mediante un botón cambiar todos los valores cambiados a los valores que por defecto trae la aplicación. De este modo mediante un solo click

podemos volver a restaurar los valores iniciales del programa en caso de que las configuraciones cambiadas no sean del agrado del usuario y no recuerde por ejemplo el estado inicial.

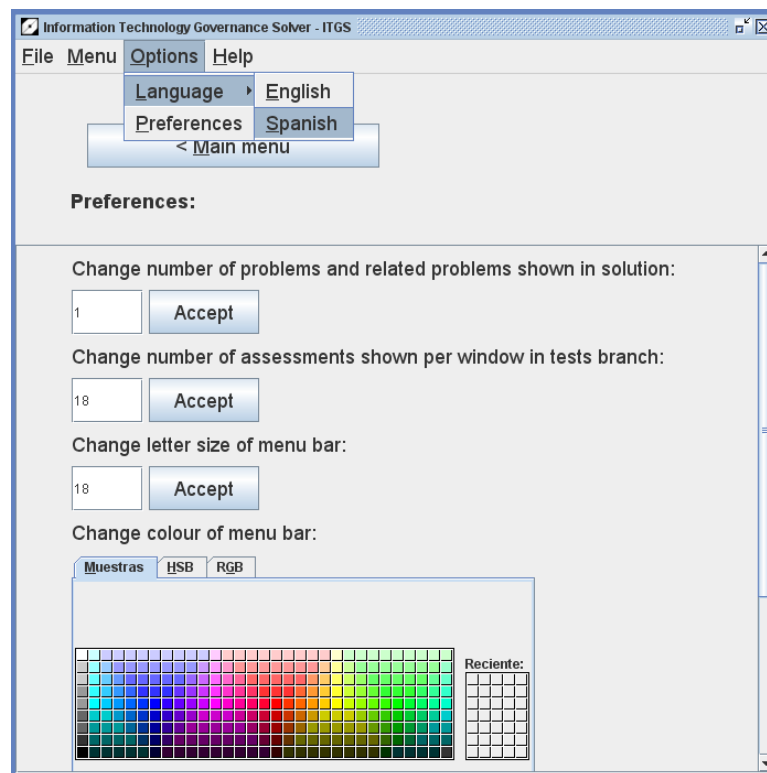


Ilustración 46. Menú de Opciones y ventana de Preferencias para personalización del usuario

Las imágenes o imágenes en movimiento y los sonidos podrían ser útiles en ciertos casos o no dependiendo de las características de la aplicación. El sonido puede ser una buena idea para ayudar a gente con problemas de visión, por ejemplo pasando el ratón por encima de botones o texto, el programa podría ir leyendo la información, en el apartado de “Líneas futuras” se habla también sobre esto, ya que en la aplicación actual no se ha implementado esta característica.

En cuanto a las imágenes, típicamente sirven para atraer la atención de un usuario y a veces transmitir las ideas mejor que las palabras. Pero en un programa meramente informativo no se debe caer en el error de llenar de imágenes la aplicación ya que más que ayudar podría entorpecer la comprensión y el uso del programa.

En la aplicación del proyecto, únicamente se han usado tres imágenes. Una es un pequeño logotipo como imagen de la aplicación, que solo aparece como icono de la aplicación y cuando salen menús emergentes del programa como ayudas o avisos. Este icono viene a sustituir el icono (la taza de café de Java) que por defecto Java aplica a sus programas, simplemente por personalización del aplicativo.



Ilustración 47. Logotipo de la aplicación ITGS



Ilustración 48. Logotipo de Java

Y las otras dos imágenes son unas interrogaciones que se usan en dos lugares de la aplicación para facilitar la comprensión de ciertos conceptos.



Ilustración 49. Logotipos de ayuda de la aplicación ITGS

En un principio se usaron imágenes dentro de los botones pero se decidió quitarlas porque al ser una pantalla de 700x700 se veía más claro usando solo texto.

En algunas partes del programa se han utilizado *tooltips* que sirven para ayudar al usuario cuando no está seguro de que función realiza algún comando. Funciona dejando un rato el cursor del ratón encima del objeto deseado sin hacer nada y un mensaje de texto saldrá aportando información.

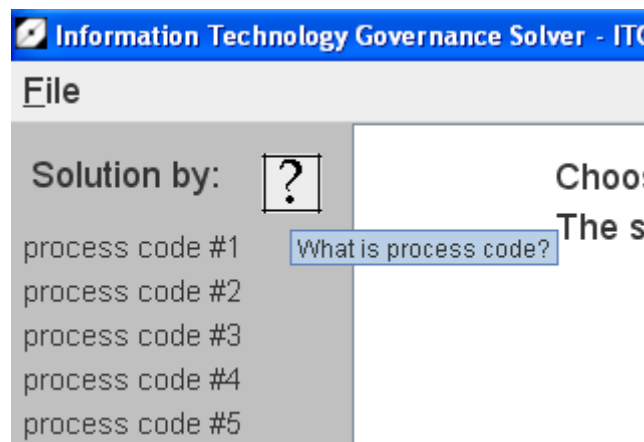


Ilustración 50. Ejemplo de tooltip. What is process code?

También se usan accesos mnemónicos para facilitar la acción de ciertas tareas, tanto en la barra menú como en los botones del programa. Mediante la pulsación de la tecla Alt + la letra que se encuentre subrayada hará el mismo efecto que si pulsamos con el ratón. A su vez los mnemónicos Ctrl + P, Ctrl + S, sirven para imprimir y guardar archivos respectivamente. La tecla Escape sirve para salir del programa o de la ventana de la solución.

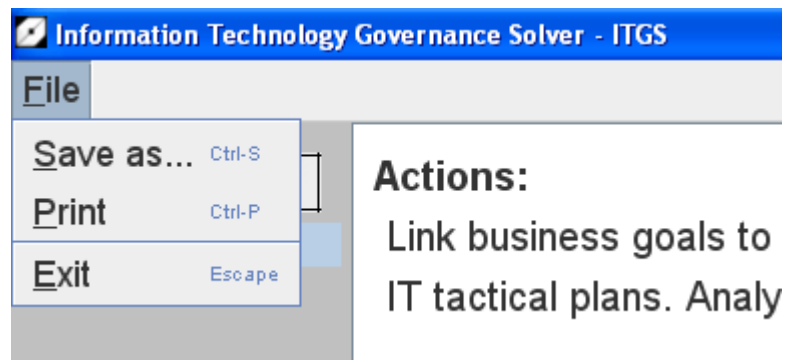


Ilustración 51. Accesos mnemónicos. (Alt + F, Alt + S, Alt + P, Alt + E, Ctrl + S, Ctrl +P, Escape)

Se permite el uso de la tecla ENTER en el caso de la ventana del administrador, tras escribir la contraseña se puede pulsar ENTER sin necesidad de hacer click con el ratón en 'Accept'. Lo mismo ocurre en la ventana de cambio de contraseña y en cambio de los datos de la conexión a la base de datos. Por otro lado la tecla ESPACIO permite ejecutar los botones que se encuentren con el foco de la ventana sin necesidad de pulsar con el ratón. El foco de un elemento viene indicado por un rectángulo alrededor de la etiqueta que tiene un botón y se puede transferir de un elemento a otro a través de la tecla Tabulador.

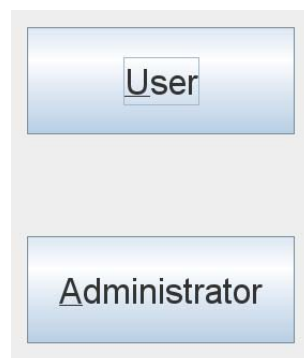


Ilustración 52. Foco en el botón User

En general el foco lo posee el elemento que se encuentre en la parte más superior de la ventana. En el caso de la consulta por dominio de COBIT, cuando hacemos click en un JRadioButton, el foco se transfiere al botón siguiente para que el usuario pueda simplemente hacer pulsación sobre la tecla ESPACIO y avanzar a la siguiente pantalla. En otras ventanas como en la de consulta por problema, el foco lo tiene la caja de texto y en ella aparece parpadeante el símbolo que indica que se puede escribir en ella, en Java es llamado el CaretPosition.

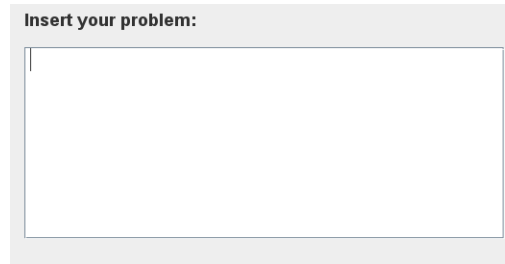


Ilustración 53. El indicador parpadeante Caret Position de Java

En la barra de menú se hace el uso de los puntos suspensivos como se recomienda en las guías de diseño cuando el submenú pulsado nos lleve a otra nueva opción o ventana, es decir que cuando se pulse no sea una acción inmediata sino que lleva a más opciones.

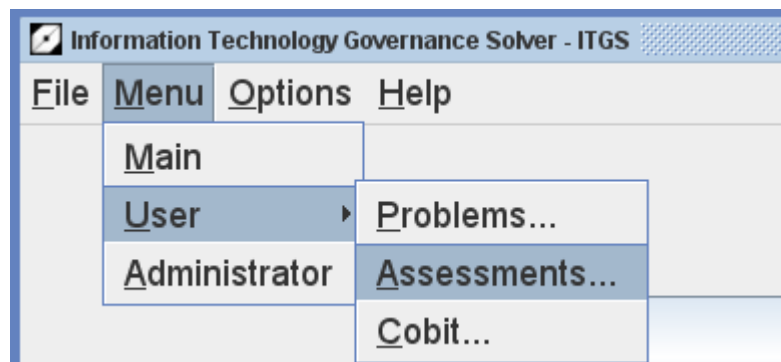


Ilustración 54. Uso de puntos suspensivos en la barra menú

En la aplicación para moverse de una ventana a otra se han implementado una serie de botones que permiten volver hacia atrás, ir hacia delante o volver a la ventana principal. Además en cualquier momento se puede hacer uso también del menú que permite ir a la ventana principal, ir a la ventana del administrador o elegir entre las tres consultas permitidas al usuario. Esto facilita la movilidad o navegación entre ventanas al usuario. En general, en la mayoría de ventanas de la aplicación se encuentra un botón para volver hacia atrás, y en algunos casos para volver al menú principal o ambas opciones a la vez. Estos botones que ayudan en la navegación, se encuentran por lo general en la parte inferior de la ventana de la aplicación, ya que se ha considerado que la información principal debería estar arriba puesto que es lo primero que se suele leer. Sin embargo, en las ventanas en las que la información es dinámica, es decir, que pueden alargarse hacia abajo la ventana y ampliarla teniendo que hacer uso de las barras de desplazamiento vertical, los botones de navegación se encuentran en la parte superior de la ventana. Esto es así porque si la información se encuentra muy hacia debajo de la ventana no tiene sentido tener que desplazarse con el scroll hasta abajo para poder navegar con comodidad, por lo que en estos casos los botones de

navegación se encuentran en la parte superior. En algunos casos además existe otro botón de navegación para moverse hacia delante.

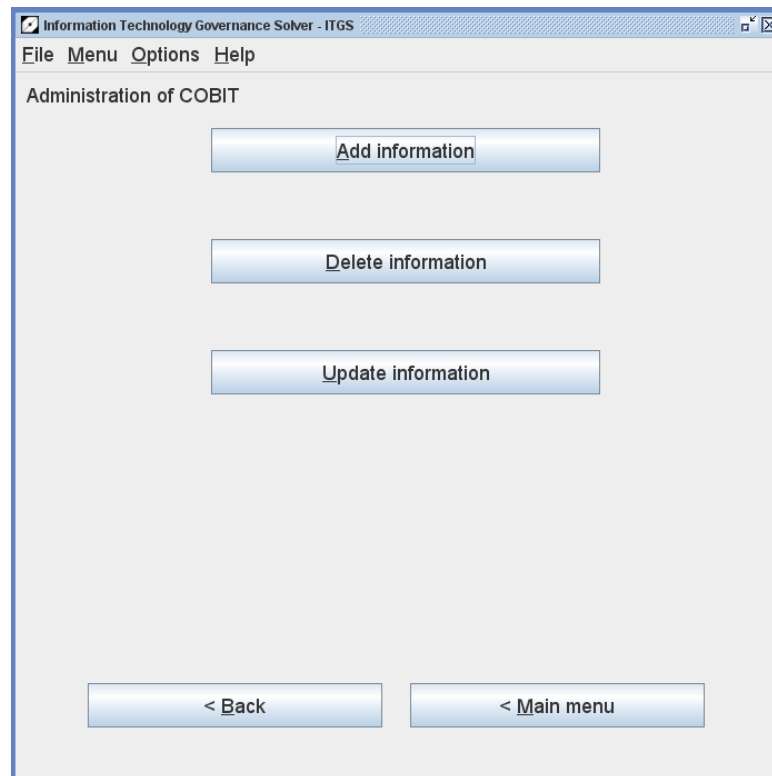


Ilustración 55. Botones de navegación Back (Atrás) y Main menu (Menú principal) en la parte inferior de la ventana

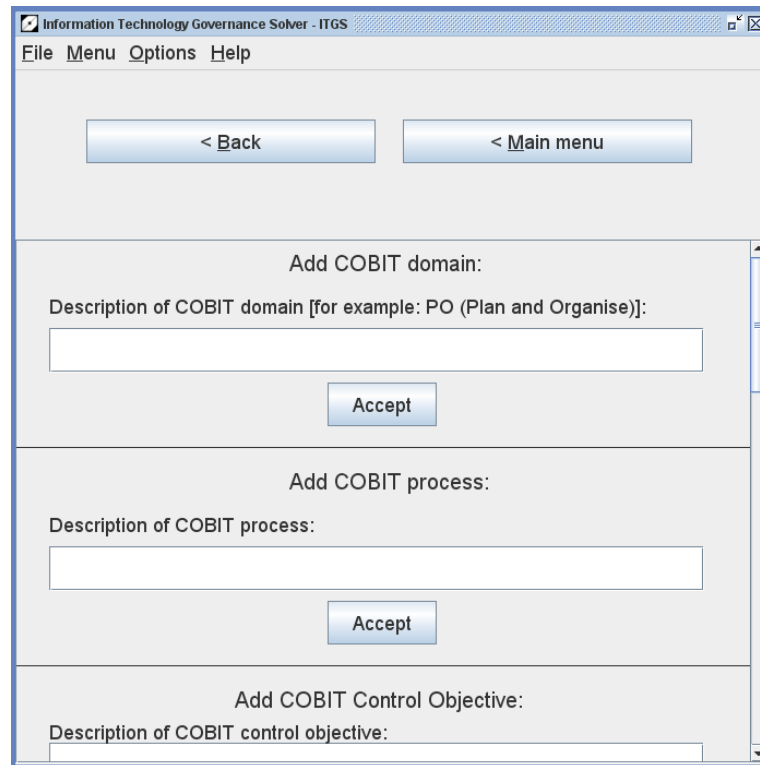


Ilustración 56. Botones de navegación Back (Atrás) y Main menu (Menú principal) en la parte superior de la ventana

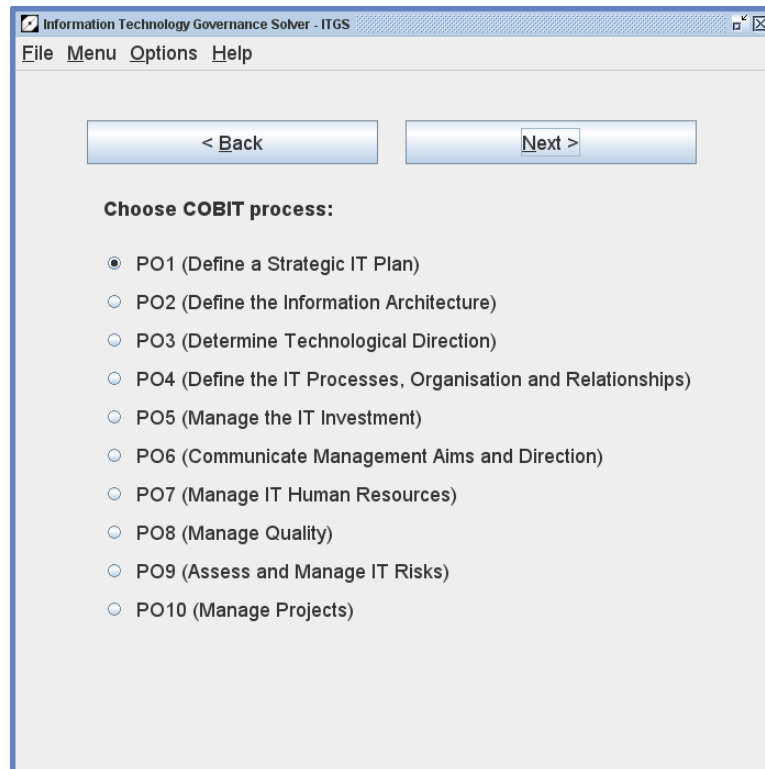


Ilustración 57. Botones de navegación Back (Atrás) y Next (Siguiente) en la parte superior de la ventana

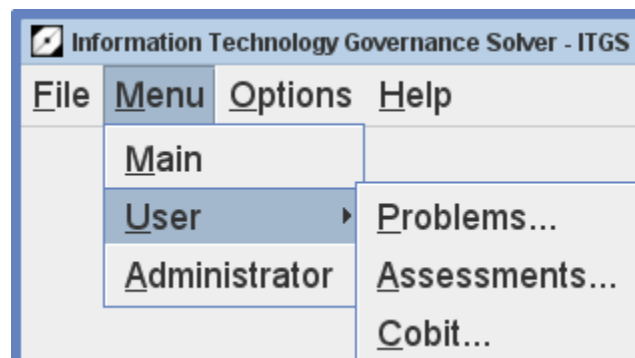


Ilustración 58. Navegación mediante la barra de menú

En algunas partes de la aplicación, más concretamente en algunas partes de la administración, en todas las bajas y en todas las modificaciones por nombre, se ha hecho el uso de una característica de Java (la variable *Point* y el componente *View*) que obtiene las coordenadas de la barra de desplazamiento vertical. En esas ventanas se realiza un refresco o actualización de la ventana para mostrar cambios en la ventana que responden a cambios que realiza el administrador, es decir que se crean ventanas de nuevo sobre la misma ventana. Ya que esas ventanas tienen scroll, los cambios pueden producirse en la parte baja de la ventana y de ser así, el refresco de la ventana haría que el scroll volviera a la parte superior, impidiendo ver la parte en la que el

administrador está realizando algún cambio, con la consiguiente molestia de tener que volver a bajar por el scroll para terminar de realizar una tarea. Esto se ha arreglado para que no pase y al realizarse un refresco no impida seguir trabajando correctamente desde la posición en la que se estaba trabajando. Esto se explica más detalladamente en la explicación del código de la administración.

Otros componentes de la aplicación aparte de los ya descritos y que tienen que ver con la interacción usuario-aplicación son los botones de acción o comando (en Java JButton), las cajas de texto (en Java JTextField), cajas de texto para contraseñas (en Java JPasswordField), las áreas de texto (en Java JTextArea), los botones radiobutton (en Java JRadioButton), los botones checkbox (en Java JCheckBox), la lista tipo árbol en la solución (en Java DefaultListModel), menús desplegables (en Java JComboBox) y etiquetas de texto (en Java JLabel). Además existen ventanas pop-up de aviso y ayuda en distintas partes del programa (en Java JOptionPane).

Los botones en cada ventana tienen el mismo tamaño ya que se recomienda el uso del mismo tamaño a ser posible para no crear confusión y con unas etiquetas lo más descriptivas y reducidas posibles de la acción que realizan. Para el botón de confirmación se usa siempre con la etiqueta 'Accept', los botones de navegación son siempre escritos como Back, Next y Main menu creando consistencia.

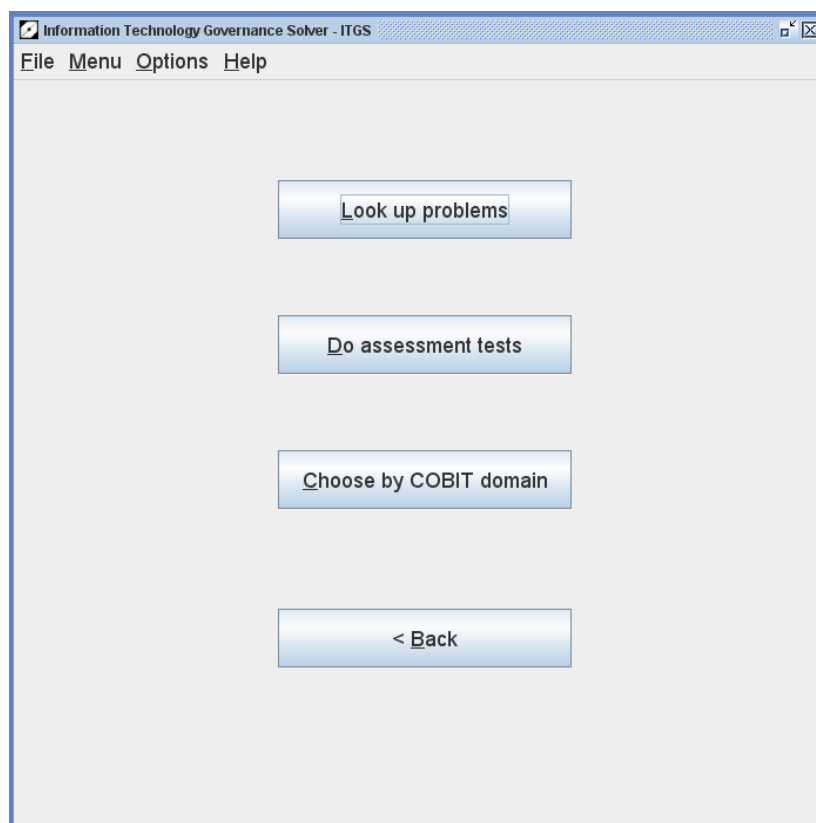


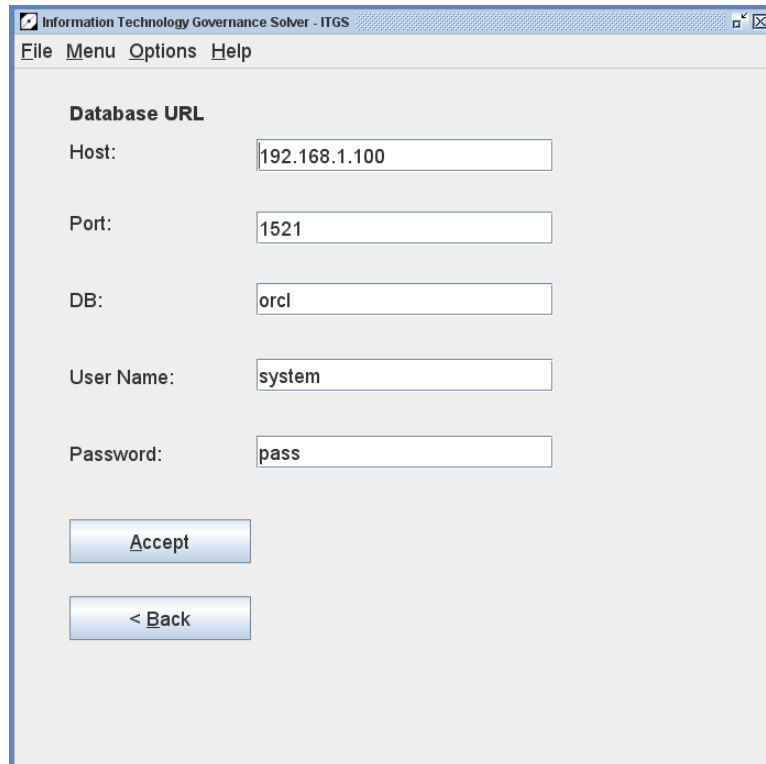
Ilustración 59. Botones de acción descriptivos y con el mismo tamaño

Las cajas de texto permiten la introducción de datos por parte del usuario y son usadas tanto en consultas como en la parte de administración. En la administración además algunas cajas de texto poseen información previa para facilitar la comprensión, por ejemplo en la administración de los parámetros de la base de datos, se muestran los datos previos guardados en fichero para consulta del administrador y permiten también su modificación.

Para las cajas de texto referentes a la contraseña y cambios de contraseña en el modo administrador se ha usado una caja de texto de tipo contraseña que Java utiliza para sustituir los caracteres introducidos por el carácter punto y evitar mostrar la contraseña como medida de seguridad.

Las áreas de texto son similares a las cajas de texto, ya que permiten la introducción de datos pero son especiales para añadir mayor cantidad de información, para añadir saltos de línea y permiten el añadido de una barra de desplazamiento para visualizar correctamente todos los datos en caso de que sobrepase el límite del área de texto. Se ha hecho además que cuando se escriba en estas áreas las palabras que lleguen al final del límite del área salten automáticamente a la siguiente línea con la palabra entera, con el fin de no tener palabras cortadas en los límites del área y se visualice mejor.

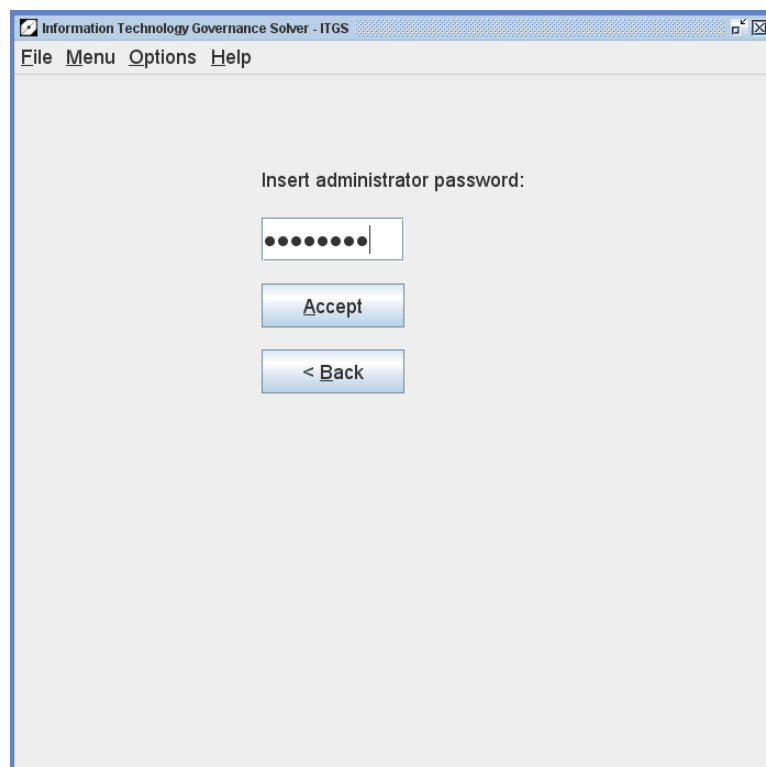
Como observación decir que todas las cajas de texto `TextField`, las áreas de texto `TextArea` y las cajas protegidas por contraseña `PasswordField`, es decir las zonas en las que se permite la introducción de caracteres tienen un límite de caracteres en concordancia con el tamaño que se ha definido en la base de datos. Es decir que si en la base de datos se permite por ejemplo una tupla con 2000 caracteres, en la caja de texto el límite sería de 2000 caracteres o similar, nunca mayor. Esto ocurre en todas y cada una de las clases de la aplicación donde existen estos componentes de introducción de datos.



The screenshot shows the 'Information Technology Governance Solver - ITGS' application window. The menu bar includes 'File', 'Menu', 'Options', and 'Help'. The main area is titled 'Database URL' and contains the following fields and buttons:

- Host:** 192.168.1.100
- Port:** 1521
- DB:** orcl
- User Name:** system
- Password:** pass
- Buttons:** 'Accept' and '< Back'.

Ilustración 60. Cajas de texto modificables con información previa de fichero



The screenshot shows the 'Information Technology Governance Solver - ITGS' application window. The menu bar includes 'File', 'Menu', 'Options', and 'Help'. The main area displays the prompt 'Insert administrator password:' followed by a password input field (masked with dots) and two buttons:

- Buttons:** 'Accept' and '< Back'.

Ilustración 61. Caja de texto protegida para contraseñas

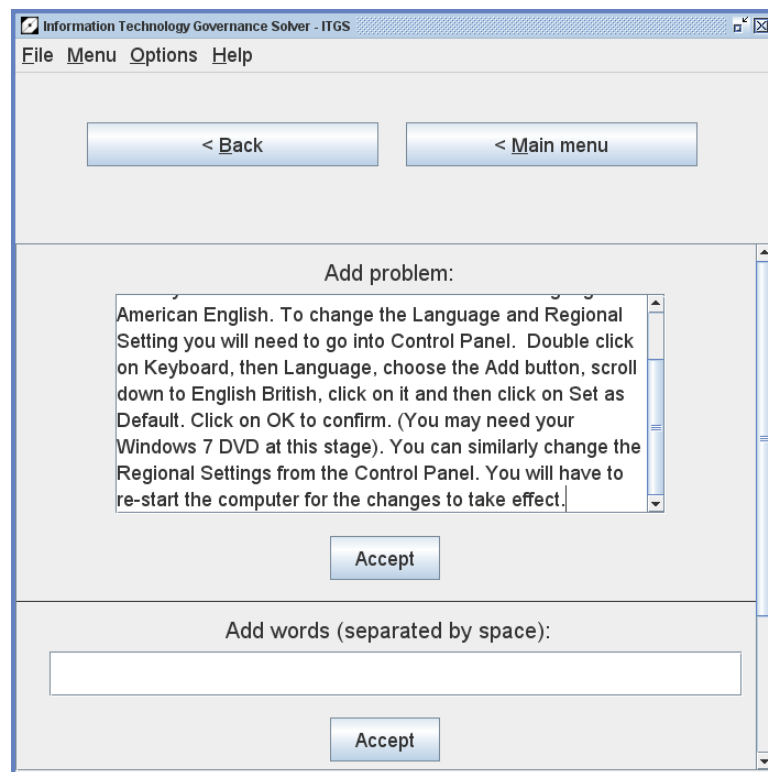


Ilustración 62. Área de texto con scroll y efecto de palabras enteras

Los botones de tipo JRadioButton sirven para elegir una opción de entre varias opciones. Se usó en las consultas por la rama de conocimiento de COBIT y en la realización de las evaluaciones por ejemplo.

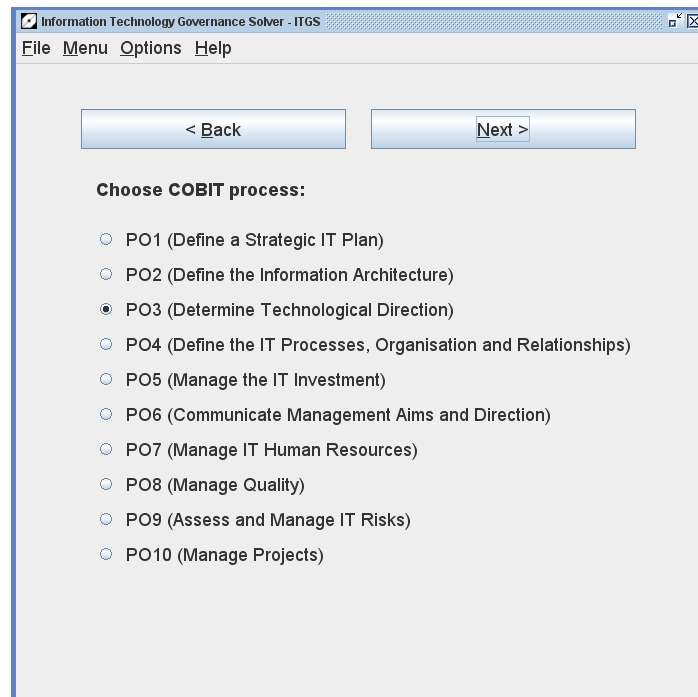


Ilustración 63. Botones JRadioButton

Y los botones tipo JCheckBox sirven para seleccionar varias opciones a la vez. Esto se ha utilizado por ejemplo en la parte de la administración para el borrado de una, varias o todas las filas (tuplas) de la base de datos de la información disponible para su modificación. Existe la posibilidad de elegir elementos uno a uno o de seleccionar o deseleccionar todos a la vez.

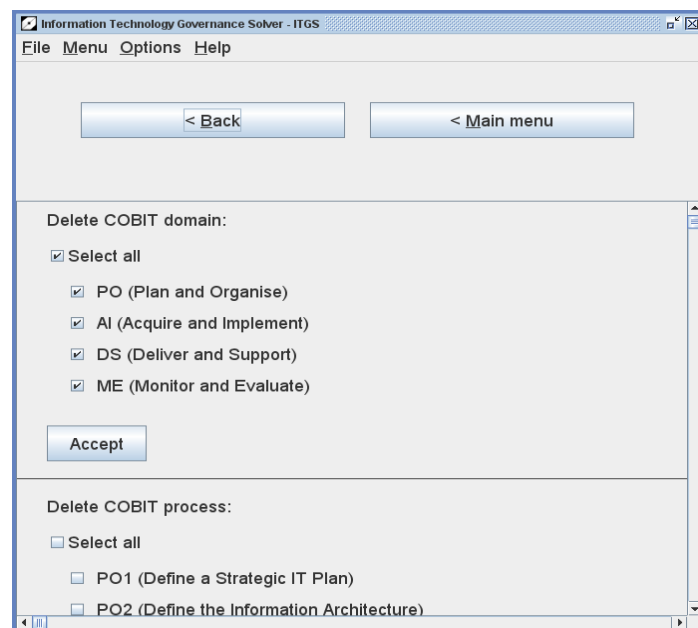


Ilustración 64. Botones JCheckBox

Ya que la solución puede venir dada por un proceso o varios de COBIT, se ha implementado una lista que permite la consulta de la solución en la misma ventana para una mayor comodidad de todos los procesos de COBIT involucrados en la solución.

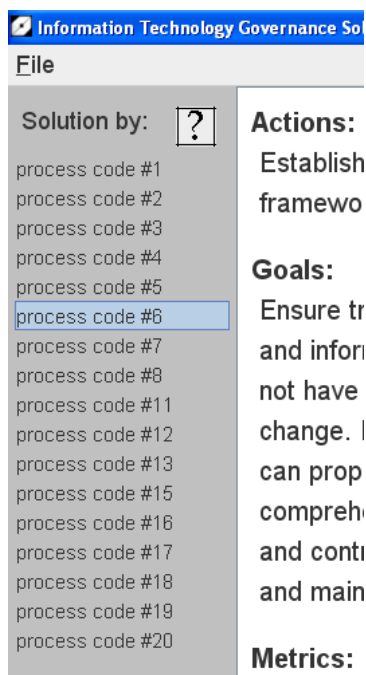


Ilustración 65. Lista de procesos COBIT para una determinada solución

El menú desplegable usado en la administración de COBIT permite usar una considerable cantidad de información de manera optimizada ya que queda recogida en un menú que ocupa poca cantidad de espacio en la ventana y que podemos abrir para consultar más información mediante el uso de barras de desplazamiento vertical y horizontal. Así se consigue reducir el espacio usado en la ventana y poder consultar la información al mismo tiempo.

Una característica añadida en esta parte de administración donde se usan estos menús es que al pulsar sobre un elemento, la caja de texto de la izquierda lee la información y la coloca en la caja de texto para facilitar la tarea de modificación de datos, ya que en caso de que el administrador solo quiera cambiar pocos caracteres porque se haya equivocado no es necesario que escriba de nuevo toda la información. También se evita el tener que escribir de nuevo una cantidad grande de información.

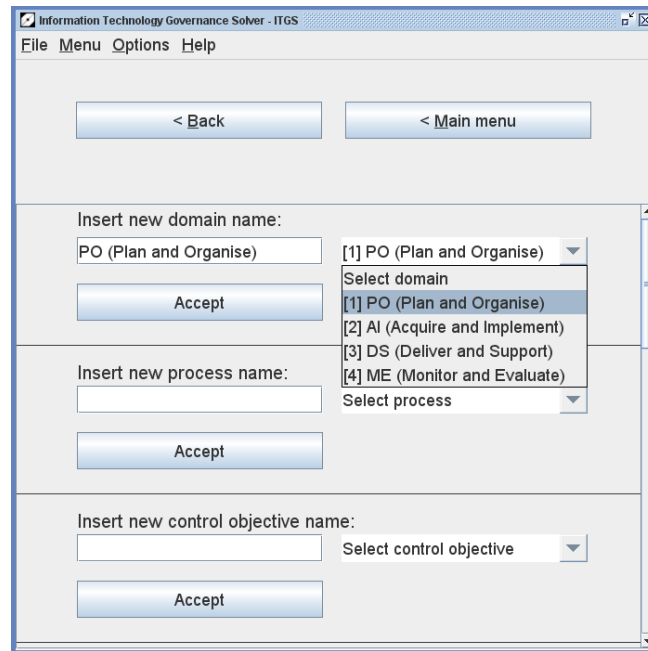


Ilustración 66. Menús desplegables

Las etiquetas de texto o JLabel se utilizan en el programa para mostrar información añadida en las ventanas, ya sea a modo de título en la ventana o a la izquierda de componentes interactivos. Por ejemplo en la ventana de cambio de contraseña se usan etiquetas al lado izquierdo, lugar recomendado para mayor comprensión de la información, de las cajas de texto para cambiar la contraseña antigua por una nueva.

Insert old password:	<input type="text"/>
Insert new password:	<input type="text"/>
Reinsert new password:	<input type="text"/>

Ilustración 67. Etiquetas en la ventana de cambio de contraseña

Por último las ventanas de aviso y ayuda muestran mensajes al usuario, son los llamados JOptionPane de Java. Ayudan a comprender partes del programa o avisan de importantes aspectos de la aplicación.



Ilustración 68. Ventana de aviso que muestra información del 'Acerca de...' de la aplicación

5.3.3 Estructura y funcionamiento del código de la aplicación

La aplicación ITGS consta principalmente de dos partes que interactúan entre sí. La propia aplicación desarrollada en lenguaje Java y la base de datos con la que se intercambia información implementada en SQL. Ambas partes son fundamentales para el correcto funcionamiento de la aplicación, pero el usuario solo utilizará la aplicación Java siendo para él la base de datos completamente abstracta. Para algunas configuraciones de la aplicación y la base de datos se hace uso de ficheros externos que se necesitan para guardar información de forma permanente.

Para la realización del código fuente se ha creado un fichero de tipo Java Class (archivos con extensión .java que al ser compilados pasarán a ser archivos .class) por cada ventana (marco o *frame* en Java) que se visualizará en la aplicación. Hay otros ficheros que no son ventanas visibles pero que también se han creado independientes de tipo Java Class para el tratamiento de algunas configuraciones como las configuraciones de la conexión de la base de datos, preferencias del usuario, etc. Dentro de cada uno de estos ficheros principales existen otras clases que están relacionadas con la clase principal que las contiene y que en algunos casos podrían estar incluidas en la clase principal como métodos en lugar de como clases pero que se han implementado como clases dentro de la clase principal para conseguir un código más legible. En general casi todos los eventos, esto es, las acciones que se producen sobre botones, otros componentes interactivos e ítems provocan eventos (en Java ActionListener, es decir escuchadores de acciones), están creados como clases dentro de las clases principales que las contienen. En algunos casos estas clases podrían haberse creado como métodos internos de las clases principales pero generarían un código menos legible. Al igual que estas clases de eventos existen otras clases, que no son de eventos, fuera de las clases principales que realizan funciones de todo tipo como por ejemplo tratamiento de cadenas de caracteres, comprobar o inicializar variables necesarias... que son llamadas desde las clases principales y se hace así por el mismo motivo de realizar un código más legible y por módulos para una posible reutilización de código. Pero fundamentalmente las clases principales se han considerado cada una de las ventanas visibles de la aplicación más las clases que anteriormente citamos de configuraciones y conexión con la base de datos. En total hay 44 archivos de tipo Java Class, esto es, 44 clases principales dentro de las cuales podemos encontrar más clases de las que hemos citado.

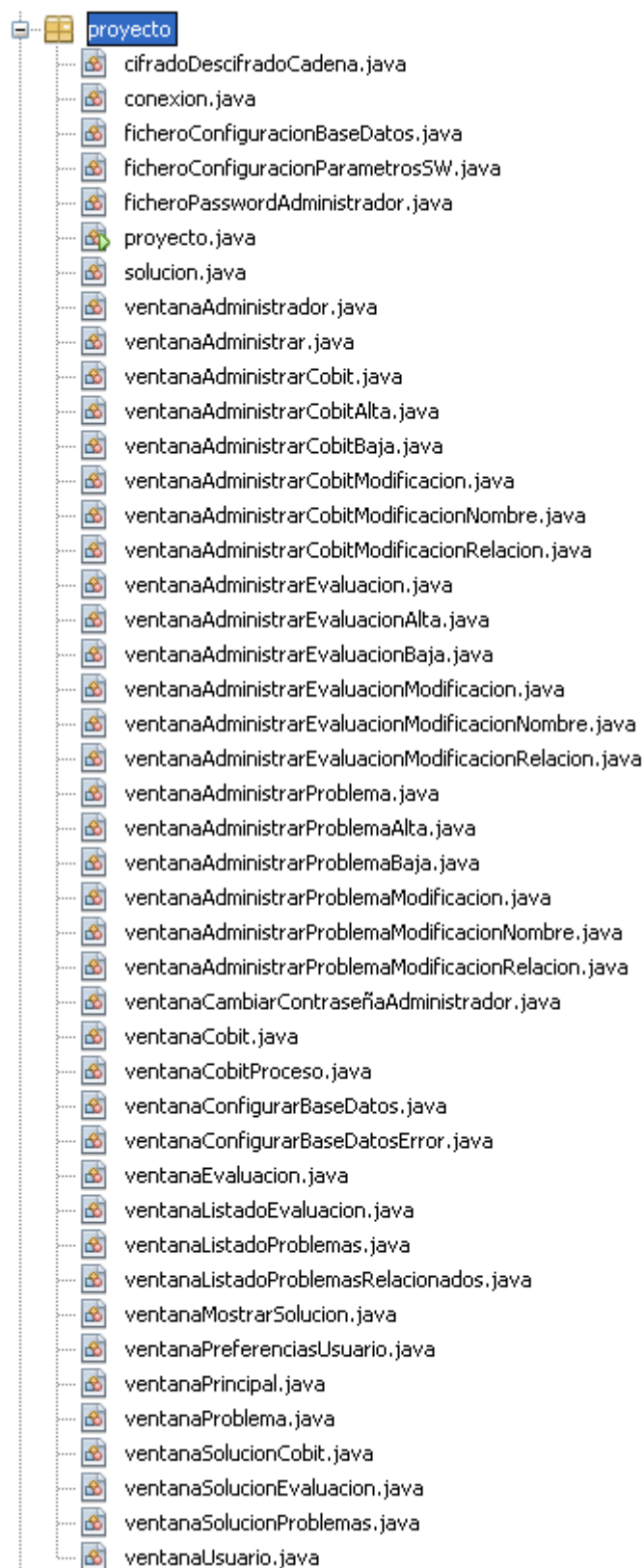


Ilustración 69. Árbol con las 44 clases principales del código fuente de la aplicación

Clase proyecto.java:

Esta es la clase que da comienzo a toda la aplicación. Contiene el método *main* del programa, es decir el método principal por el que Java empieza a ejecutar la aplicación. Esta clase realiza automáticamente una serie de llamadas a otras clases entre las que encontramos a las clases:

- *ficheroConfiguracionBaseDatos.java*
- *ficheroPasswordAdministrador.java*
- *ficheroConfiguracionParametrosSW.java*
- *conexion.java*
- *ventanaPrincipal.java*

Las acciones que se realizan tras el llamado a esas clases son la de crear, en el caso de que no existan, el directorio 'ficheros' en el que todos los ficheros serán guardados, el fichero de configuración de la base de datos (*ficheroConfiguracionBaseDatos.txt*), el fichero que contiene la contraseña del administrador (*ficheroPasswordAdministrador.txt*) y el fichero de configuración de parámetros de la aplicación (*ficheroConfiguracionParametrosSW.txt*). Es decir se crean los tres ficheros que la aplicación utiliza para guardar configuraciones del programa, configuración de la conexión de la aplicación con la base de datos y la contraseña del administrador. En el caso de que se borrarán accidentalmente alguno de estos ficheros, la aplicación los crearía de nuevo con unos valores por defecto. Las clases encargadas de esto son:

ficheroConfiguracionBaseDatos.java: es la encargada del tratamiento del fichero de configuración de la base de datos para su conexión.

ficheroPasswordAdministrador.java: se encarga del fichero que guarda la contraseña del administrador.

ficheroConfiguracionParametrosSW.java: guarda la configuración de parámetros de personalización de la aplicación (*SW=Software*) como el idioma, color, tamaño letra del menú, número de problemas mostrados y número de evaluaciones mostradas por pantalla.

Estas tres clases son muy similares internamente entre sí, ya que realizan tareas típicas de tratamiento de ficheros, únicamente cambiando la información que hay en cada una de ellas. Todas comprueban si el fichero está creado o no, si está creado no hace nada y si no está creado lo crea con unos valores por defecto, y tienen las funciones de lectura, guardado y borrado típicas de fichero. Además dentro de las clases de configuración de la base de datos y de la contraseña del administrador se hace una llamada a la clase ***cifradoDescifradoCadena.java*** que se encarga de descifrar la lectura de datos o cifrar la escritura de datos por medio de un cifrado de sustitución monoalfabeto. De este modo, cada vez que una clase modifique datos en uno de estos ficheros, automáticamente serán cifrados (o descifrados para su lectura). El cifrado sigue la función $C(x)=(x+1) \bmod N$, y el descifrado $D(x)=(x-1) \bmod N$, donde x es el

carácter a cifrar o descifrar, N la longitud del alfabeto y C y D son Cifrado y Descifrado respectivamente. Es decir que para cifrar se le añade el siguiente carácter ASCII al original y para descifrar se le resta un carácter ASCII al original. Por ejemplo la contraseña original “password” al cifrarla se convertiría en “qbttxpse”. En el fichero de configuración de parámetros de la aplicación no se realiza un cifrado ni por tanto descifrado ya que no es información crítica de seguridad.

A continuación se comprueba el estado de la conexión con la base de datos mediante la clase **conexion.java**. Esta clase hace una llamada a la clase **ficheroConfiguracionBaseDatos.java** para leer los datos de la conexión desde el fichero e intenta conectarse a la base de datos, en caso positivo la aplicación podrá ejecutarse sin problemas, en caso negativo el programa nos avisará cuando lo vayamos a usar de que existe un problema. Para realizar la conexión se hace necesario el uso de la librería *ojdbc6* que permite la conexión y ejecución de operaciones sobre la base de datos desde el lenguaje Java. Dentro del método que establece la conexión correcta, se realiza una llamada a otro método de la clase conexión para aumentar el número de cursores que la base de datos utiliza. El cursor en una base de datos se usa para recorrer las filas de cada consulta. Existen ciertas partes del programa donde se realizan gran cantidad de consultas a la base de datos (especialmente en la administración) y si se hacen cambios en el programa en esas partes, el programa hace más consultas internas. Esto puede provocar que el número de cursores abiertos en la base de datos supere el límite establecido por defecto y provoque un error en la base de datos (*Exception ORA-01000: número máximo de cursores abiertos excedido*) y como consecuencia un fallo en el programa. El número por defecto suele estar entre 50 y 300 cursores abiertos, aunque cada sistema operativo determina el valor apropiado, este número puede ser a veces insuficiente. Para evitar este problema de la base de datos, cada vez que la aplicación se conecta a la base de datos se incrementa automáticamente el valor del número de cursores que se pueden abrir al mismo tiempo con un valor de 65000 cursores (el valor máximo es 65535). Además de esto, en algunas partes problemáticas, se han cerrado las conexiones que abren los cursores cada vez que se realiza una instrucción *resultSet*, que es la que devuelve un puntero (cursor) sobre la fila de la tabla de la base de datos para que se pueda recorrer de principio a fin.

Hay que decir que en los momentos en los que se sabe que el programa puede producir algún error, como en el caso de la conexión a la base de datos, se han tratado para mostrar con avisos o ayudas al usuario del posible problema y que esto no haga funcionar mal a la aplicación para que pueda seguir siendo operativa dentro de lo posible, es decir que se intenta hacer del programa un sistema tolerante a fallos.

Por último se da paso a la creación de la ventana principal por medio de la clase **ventanaPrincipal.java** que será la primera ventana visible al usuario y también al administrador, es decir la ventana o *frame* principal del programa listo para poder usarse. Pero antes de mostrar la ventana de la aplicación, se muestra por pantalla un mensaje de aviso sobre el Copyright de la aplicación. Este mensaje se cierra

automáticamente a los 4 segundos, o si no se quiere esperar, se puede pulsar sobre su botón “Accept” o sobre el botón X de cierre. Tras los 4 segundos o tras aceptar o cerrar el mensaje, aparecerá la ventana principal.

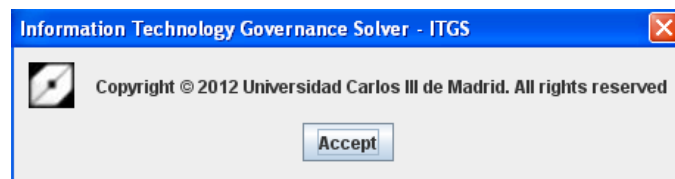


Ilustración 70. Mensaje del copyright de la aplicación al inicio

Clase ventanaPrincipal.java:

Una de las primeras acciones que se realizan en esta clase de la ventana principal es la de leer los datos del fichero de configuración de parámetros del programa. Esto se hace realizando una llamada a la clase **ficheroConfiguracionParametrosSW.java** para obtener los parámetros por defecto de la aplicación o los parámetros que el usuario guardó en su personalización. Desde este fichero la aplicación será ejecutada en un determinado idioma (inglés o español), con un tamaño de letra y un color en la barra menú. Además se crea la ventana visible al usuario o administrador con un tamaño de 700x700 píxeles, centrada, colocada en el centro del monitor, y se inhabilita el botón de maximizar la ventana.

Desde la ventana principal del programa se puede navegar hacia casi cualquier parte del aplicativo. Proporciona menús y botones para poder usar las principales funciones del programa desde un principio.

La barra menú consta de los menús *File* (Archivo), *Menu* (Menú), *Options* (Opciones) y *Help* (Ayuda). El menú *File* consta de los ítems “Save as...”, “Print” y “Exit”. Los ítems “Save as...” (Guardar como, para guardar en archivo la solución) y “Print” (Imprimir, para imprimir la solución) solo están habilitados en la ventana de la solución del programa, ya que es el único lugar donde estas acciones cobran sentido, cuya barra de menú solo consta del menú File e incluye el ítem “Exit” (Salir, sale del programa).

El menú *Menu* consta de dos ítems “Main” y “Administrator” y un submenú que es “User”. El submenú “User” está formado por tres ítems que son “Problems...”, “Assessments...” y “Cobit...”. *Menu* sirve para ayudar en la navegación del programa al usuario, desde él podemos acceder a la ventana principal (Main), a la ventana del administrador (Administrator) y desde el submenú “User” podemos elegir uno de los tres ítems de consulta del problema del usuario. Estos tres ítems permiten consultar por la descripción del problema escrito (ítem Problems...), consultar mediante la realización de evaluaciones (ítem Assessments...) o consultar mediante el conocimiento del modelo COBIT 4.1 (ítem Cobit...).

El menú *Options* (Opciones) consta de un submenú “Language” (Idioma) con dos ítems “English” (Inglés) y “Spanish” (Español). Según se elija un idioma u otro, la aplicación cambiará el idioma elegido y guardará el nuevo idioma en el fichero de configuraciones llamado *ficheroConfiguracionParametrosSW.txt*. Para configurarse, la aplicación necesita comenzar desde la ventana principal (llamada a la clase ***ventanaPrincipal.java***) para aplicar los cambios correctamente. Se necesita volver a crear la ventana principal para el nuevo idioma ya que se cambia el menú por uno nuevo en otro idioma, lo cual necesita la destrucción del anterior *frame* por uno nuevo, y además se necesita conocer en qué ventana se encuentra el usuario o administrador, por lo que lo más óptimo es comenzar de nuevo desde la ventana principal para que a partir de ella, el resto de ventanas se creen con el idioma correcto. El lenguaje español no está implementado en todas las ventanas de la aplicación, solo en unas pocas a modo de demostración y el funcionamiento en las ventanas en las que está disponible es que se lee desde el fichero de configuraciones el idioma que esté guardado, y cuando se conoce el idioma usado, las distintas ventanas crean sus componentes en un idioma u otro por medio de sentencias *if*. Se habla más sobre el idioma en el apartado de “Líneas futuras”.

El menú *Options* está formado además por un ítem llamado “Preferences” (Preferencias), desde donde se pueden cambiar ciertas configuraciones para personalización del programa por parte del usuario, mediante la clase ***ventanaPreferenciasUsuario.java***. Esta clase se explicará más adelante.

Por último el menú *Help* (Ayuda) está compuesto por dos ítems, “User guide” (Manual de usuario) y “About” (Acerca de). El menú *Help* ofrece la asistencia mediante una guía o manual de usuario, mediante la ejecución de un archivo previamente creado en formato PDF desde el ítem “User Guide”, y muestra un mensaje con información reducida sobre la aplicación, así como de los creadores y un *email* de contacto en el ítem “About”.

En resumen, desde la barra del menú, que aparece en todas las ventanas de la aplicación, no solo en la ventana principal sino en todas, con la excepción de que en la ventana de la solución aparece reducida con solo el menú File, se puede navegar desde cualquier punto del programa hacia una nueva consulta por una rama distinta de consulta o cambiar de perfil de usuario (usuario o administrador), también desde cualquier punto es posible cambiar las preferencias de personalización, acudir al manual de usuario o al acerca de la aplicación y salir de ella. En la ventana de la solución además se puede guardar, imprimir y salir como ya se ha indicado.

Desde los botones de las ventanas de la aplicación se puede llegar a las mismas opciones que la barra de menú proporciona en el menú Menu, sin embargo, las opciones existentes en el menú Options y en el menú Help, así como las opciones de guardar e imprimir disponibles en el menú File de la ventana solución, solo se pueden llegar a ellas desde la propia barra de menú, bien mediante el uso del ratón o con accesos mnemónicos.

A continuación de la barra de menú, la ventana consta de dos botones (User y Administrator) con los que se puede observar que existen dos perfiles de usuario: El “usuario” (el cliente), y el “administrador”.

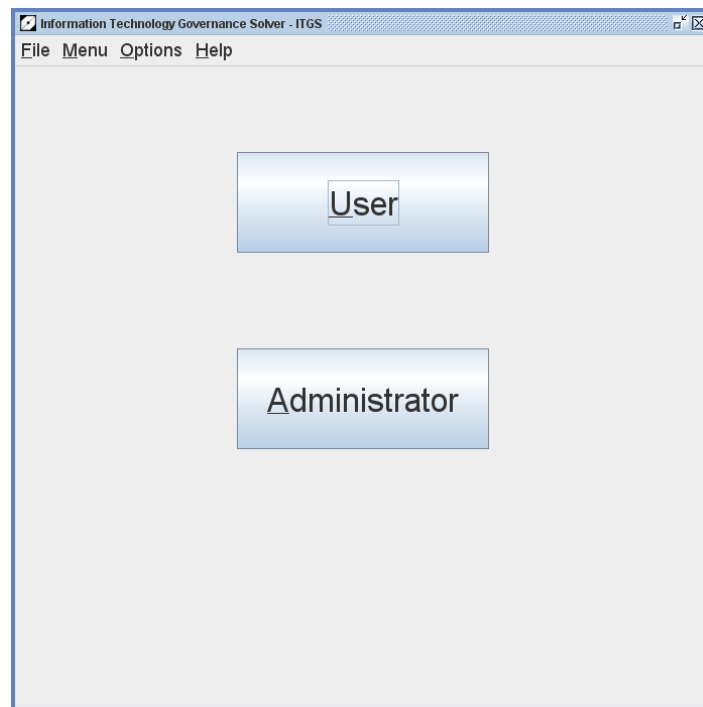


Ilustración 71. Ventana “Principal” de la aplicación con los perfiles “Usuario” y “Administrador”

Al pulsar sobre el botón “User” se producirá un evento provocando una llamada a la clase **ventanaUsuario.java** y si se pulsa sobre el botón “Administrator” la llamada se producirá hacia la clase **ventanaAdministrador.java**.

Clase ventanaUsuario.java:

La ventana usuario es la ventana de las consultas que el usuario normal, es decir, el usuario no administrador del sistema, puede realizar, para que se entienda es el cliente de la aplicación.

El sistema posee tres modalidades o ramas en la búsqueda de la solución a un problema TI dado por el usuario.

La búsqueda por una de las ramas es una búsqueda exhaustiva o de fuerza bruta en la base de datos, el sistema recibe directamente a través de una frase descriptiva el problema TI del usuario y busca su solución.

Otra rama de consulta es la modalidad de las evaluaciones, cuestionarios o assessments, en la que el usuario realiza una serie de cuestionarios para evaluar la situación de su empresa de cara a posibles problemas de TI y conocer la madurez de la compañía en sus distintos procesos de COBIT.

En la última modalidad o rama, el usuario debe ir eligiendo por áreas del gobierno TI, es decir por dominios y procesos TI pertenecientes a COBIT, eligiendo las áreas más cercanas a su problema. En esta rama se supone que el usuario debe tener un mínimo de conocimiento sobre temas relacionados con las TI y en particular estar algo familiarizado con la metodología COBIT.



Ilustración 72. Las tres ramas posibles de consulta del usuario en la ventana "Usuario"

La ventana usuario consta de tres botones por cada rama disponible de consulta. El primer botón llamado "Look up problems" es decir consultar problemas, coincide con la primera rama explicada de introducir una frase descriptiva del problema TI. El siguiente botón es "Do assessment tests" esto es hacer los tests de evaluación, que corresponden a la rama de los cuestionarios. El tercer botón es "Choose by COBIT domain" que trata sobre la tercera rama correspondiente a la consulta por área de dominio de COBIT. El último botón "Back" (Atrás) ayuda al usuario a volver a la ventana anterior, que sería la de la ventana principal, para facilitar la navegación por la aplicación. De cada botón pulsado se generaría un evento creando una ventana nueva para la rama seleccionada.

A partir de ahora se usará la siguiente nomenclatura para cada rama:

La primera rama es la de “Rama de búsqueda”, la segunda “Rama de evaluación” y la tercera “Rama de COBIT”.

Rama de COBIT:

La clase **ventanaCobit.java** es llamada desde la clase **ventanaUsuario.java** cuando se ha seleccionado el botón “Choose by COBIT domain” en la ventana usuario. Desde esta nueva ventana se puede realizar una consulta eligiendo por áreas de COBIT, primero por un área de dominio de COBIT y luego por un área de proceso de COBIT.

Pero antes, se comprueba que existan dominios en la base de datos, ya que si no existen no tiene sentido continuar y en este caso el sistema mostrará un aviso de la no existencia de datos. En caso de existir dominios el usuario podrá elegir un dominio de entre los dominios que tiene COBIT y continuar para elegir un proceso dentro de ese dominio o volver hacia atrás, en cuyo caso volvería a la clase **ventanaUsuario.java**.

En esta rama el usuario elige por conocimiento propio las áreas afectadas de su problema, se comienza eligiendo por un dominio de alcance global (basado en los dominios de COBIT: Planificación y Organización, Adquisición e Implantación, Entrega y Soporte, Monitorización y Evaluación). Los dominios son leídos desde la base de datos y representados en la ventana mediante JRadioButtons que permiten la selección de un único dominio cada vez. Posteriormente si hacemos *click* en “Next” (Siguiente) el sistema da a elegir entre varios procesos pertenecientes al dominio elegido por el usuario. Al hacer *click* sobre “Next” en la **ventanaCobit.java** se creará la clase **ventanaCobitProceso.java** que será la que permita elegir los procesos que estén relacionados con el dominio elegido en la ventana anterior. Los procesos son leídos desde la base de datos y representados mediante JRadioButtons que permiten la selección de un único proceso cada vez. En cambio si se pulsa sobre el botón “Back” en lugar de en “Next” el usuario regresaría a la ventana anterior que sería la ventana usuario.



Ilustración 73. Ventana Cobit (Dominio) con los cuatro dominios de COBIT 4.1

Como detalle para ayudar al usuario, el foco nada más entrar en la ventana para elegir el dominio lo tiene el botón “Back” por si se equivocó y desea volver hacia atrás rápidamente mediante el teclado (o ratón), pero en cuanto elijamos un dominio (que es un JRadioButton ya que solo se permite elegir uno cada vez) el foco se le entregará al botón “Next” para que el usuario pueda hacer uso del teclado pulsando ‘Espacio’ si quiere, facilitándole la tarea de navegabilidad, o puede hacer uso del ratón también. Si pulsamos “Next” sin elegir dominio el programa nos dirá que pulsemos sobre algún dominio para continuar. Si el dominio elegido no tiene ningún proceso relacionado el sistema nos lo indicará y nos pedirá que elijamos un dominio con proceso asociado para poder continuar. Por defecto la aplicación usa la información completa de dominios y procesos de COBIT los cuales están relacionados, pero como se verá más adelante en las tareas de administración podría ocurrir que se introdujeran datos no relacionados entre sí.

Como hemos dicho, al pulsar sobre “Next” habiendo elegido un dominio de COBIT la clase **ventanaCobitProceso.java** hace aparición permitiendo elegir un proceso de los varios procesos que tenga el dominio seleccionado en la ventana anterior. En esta ventana podemos volver hacia atrás mediante el botón “Back”, es decir volveríamos a la ventana de elegir dominio de COBIT, o pulsar sobre “Next” lo cual nos llevaría a una nueva ventana. En esta ventana de elección de proceso ocurre lo mismo que en la anterior sobre el foco para ayudar al usuario en la navegabilidad al pulsar sobre un proceso. Si el proceso seleccionado no tiene una solución relacionada, el sistema avisará indicando que se seleccione un proceso que tenga solución. Un proceso tiene solución cuando está relacionado con un objetivo de control (el objetivo de control a su vez está relacionado con una serie de acciones, métricas, metas y otros objetivos que forman parte de la solución). Por defecto el programa tiene asociada una solución a cada proceso de COBIT, pero como ocurría antes, si se introducen datos en la administración y no se relacionan correctamente podría pasar que no obtengamos la solución. Esto se explicará en la parte de administración.

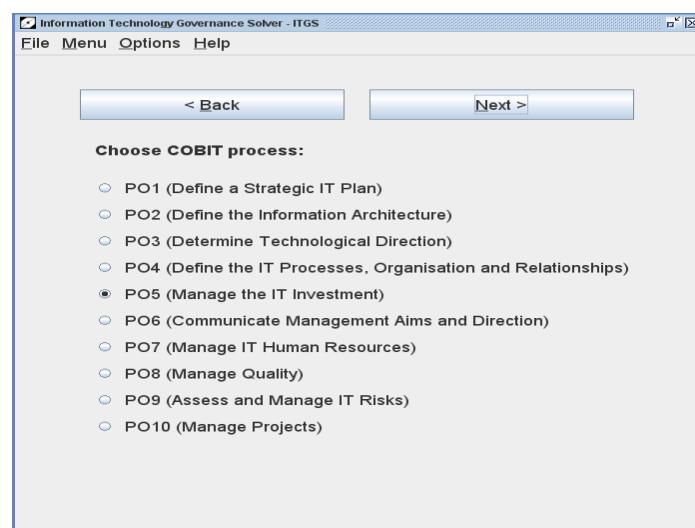


Ilustración 74. Ventana Cobit Proceso con los diez procesos relacionados con el dominio PO de COBIT

4.1

En ambas ventanas de elegir dominio y proceso, si el administrador introdujera más información, el sistema crearía una barra de desplazamiento tanto horizontal como vertical si fuera necesario para visualizar correctamente toda la información.

Una vez elegido el proceso deseado y se pulse el botón “Next”, la aplicación realizará una llamada a la clase **ventanaSolucionCobit.java** y a la clase **solucion.java** y mostrará un aviso por pantalla de que la solución se mostrará en una nueva ventana (un nuevo frame independiente de la ventana principal de la aplicación) a la vez que aparecerá una nueva ventana en el frame de la aplicación.

La clase **solucion.java** es la nueva ventana independiente que aparece tras aceptar el mensaje de que la solución será mostrada. Esta clase se explicará más adelante ya que está relacionada con otras clases. Pero básicamente es la encargada de, junto con la clase **ventanaMostrarSolucion.java**, mostrar la solución por medio de toda la información disponible de COBIT a la consulta realizada por el usuario.

La clase **ventanaSolucionCobit.java** es la otra ventana que aparece pero en el frame de la aplicación. Esta ventana consta de tres botones que nos permiten consultar problemas de la base de datos que estén relacionados con nuestra elección de dominio y proceso (botón “Look up related problems”), así como realizar una nueva consulta por área de dominio de COBIT, es decir, comenzar de nuevo con una nueva selección por área de dominio y proceso (botón “New consult by COBIT domain”), o bien volver al menú principal (botón “< Main menu”).

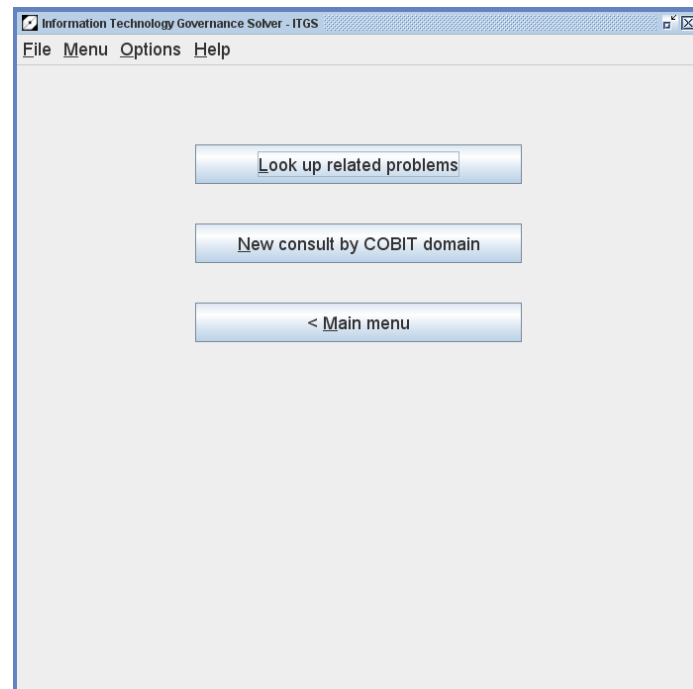


Ilustración 75. Clase y ventana VentanaSolucionCobit.java

Si se pulsa sobre el botón de menú principal se realiza una llamada al evento del botón para que esconda la actual ventana y muestre la ventana principal, **ventanaPrincipal.java**.

Si se pulsa sobre el botón de hacer nueva consulta por dominio de COBIT, se hace una llamada al evento del botón que esconde la ventana actual y muestra la **ventanaCobit.java** que es la que da comienzo a la consulta por esta rama de COBIT.

Si se pulsa sobre el botón de consultar problemas que estén relacionados, se hace una llamada al evento del botón que esconde la ventana actual y crea la clase **ventanaListadoProblemasRelacionados.java**, la cual muestra los problemas encontrados en la base de datos que estén relacionados con la consulta por la rama COBIT. Justo antes de llamar a esta clase se realiza una llamada a la clase **solucion.java** que se encarga de obtener los problemas de la base de datos que estén relacionados con un determinado proceso. La clase muestra el número de problemas que el fichero *ficheroConfiguracionParametrosSW.txt* tenga como valor, por defecto el valor es 4, pero el usuario puede cambiarlo según se explica en la clase **ventanaPreferenciasUsuario.java**. Después se crea la ventana con el listado de problemas y en caso necesario con una barra de desplazamiento vertical y horizontal, y consta de dos botones en la parte superior para volver hacia atrás mediante el botón “Back” (volveríamos a la ventana **ventanaCobitSolucion.java**) o para continuar cuando se seleccione un problema del listado con el botón “Next”. Los problemas están asociados a JRadioButtons que permiten la selección de un problema cada vez. Al pulsar sobre un problema el foco se transferirá al botón Siguiente y al pulsar sobre este se creará un nuevo frame independiente, una nueva ventana, con la solución de COBIT para ese problema, esto se consigue mediante una llamada a la clase **solucion.java**. Si no se elige un problema y se pulsa “Next” la aplicación avisará de que se debe elegir un problema para continuar.

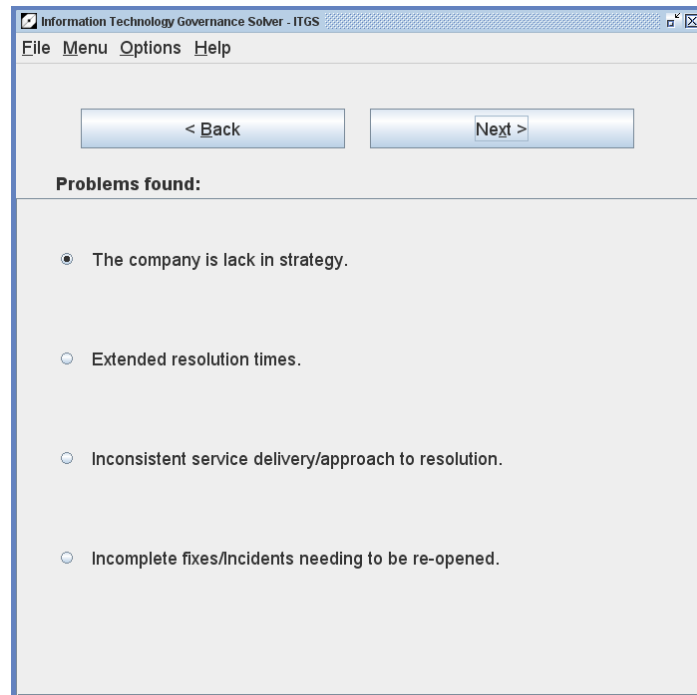


Ilustración 76. Listado de problemas relacionados en la clase ventanaListadoProblemasRelacionados.java desde la rama de COBIT

Como observación, si ya se tenía creada una ventana de solución y ahora ejecutamos un botón que crea otra nueva ventana de solución, la anterior ventana de solución se destruirá para evitar casos de interacción de eventos entre ventanas. Es decir, solo se puede tener abierta una ventana de solución, no varias a la vez. Pero se puede consultar y exportar a fichero, y a continuación cerrarla o no ya que la aplicación lo hace automáticamente, para consultar una nueva solución.

Clase solucion.java:

La clase ***solucion.java*** es la clase encargada de obtener la solución de la base de datos. La solución viene dada por un conjunto de información que proporciona COBIT 4.1 cuyos datos son acciones, metas, métricas, objetivos de control, objetivos TI y objetivos de negocio. La solución siempre está relacionada con un proceso de COBIT, y a partir de un proceso de COBIT se puede obtener toda la información de COBIT. Por tanto como en COBIT 4.1 hay 34 procesos en total, habrá un total de 34 soluciones disponibles, a no ser que en el futuro se añadan más procesos en la administración. Para obtener toda esta información la clase *solucion* tiene unos métodos necesarios para obtener procesos, problemas y otras características que se describen a continuación. A esta clase se llega por cualquiera de las tres ramas de consulta disponibles para el usuario. La clase *solucion* está formada por los siguientes métodos:

dialogoMostrarSolucionEnBreve: Este método muestra un mensaje de que la solución será mostrada en una nueva ventana en breve. Esto ocurre cuando en una ventana de consulta por las tres ramas disponibles se llega a la última ventana tras la cual va a aparecer la ventana de la solución.

obtenerProblema: Obtiene un problema a partir de un proceso de COBIT. A este método se le puede enviar un proceso de COBIT y devolverá una lista de problemas que estén relacionados con el proceso en la base de datos y un booleano indicando si se encontró o no un problema.

obtenerProcesoCobit: Obtiene un proceso de COBIT a partir de un problema dado. El método puede recibir un problema y consultará en la base de datos para obtener un listado de procesos de COBIT que están relacionados con dicho problema.

obtenerArbol: Obtiene una lista de procesos que serán los que proporcionarán la solución, puesto que la solución es dada para cada proceso. Además crea la ventana o frame que la clase ***ventanaMostrarSolucion.java*** utilizará posteriormente para mostrarla con información, es decir, crea la ventana independiente de la ventana de la aplicación que muestra la solución. El nuevo frame, a diferencia de la ventana de la aplicación, es maximizable por completo, para facilitar la lectura del texto de la solución, teniendo habilitados el botón de maximizar, además del de minimizar y cerrar y aparecerá maximizado desde el principio. Si se decide restaurar la ventana su tamaño será de 500x500 píxeles siendo más pequeña que la ventana de la aplicación cuyo tamaño máximo es de 700x700 píxeles, permitiendo ver ambas ventanas sin que se solapen.

obtenerSolucion: A este método se le envía una lista de procesos y se realiza una llamada al método *obtenerArbol*, el cual, crea la estructura de una lista tipo menú con los procesos y la ventana física o frame de la ventana independiente de la solución. Después realiza una llamada al método *obtenerSolucionProceso*.

obtenerSolucionProceso: Obtiene a partir de un proceso dado la solución de COBIT consultando toda la información en la base de datos formada por la siguiente información: acciones, metas, métricas, objetivos de control, objetivos TI y objetivos de negocio. A partir de un proceso se puede obtener toda la información de COBIT ya que está relacionado con todas las tablas y con las tablas que no tienen relación directa se pueden obtener a través de otras tablas directas con la tabla proceso. Con toda la información recogida realiza una llamada a la clase ***ventanaMostrarSolucion.java***.

Clase ventanaMostrarSolucion.java:

Esta clase es llamada con el conjunto de información (que es la solución final) obtenida de COBIT a través de la base de datos desde la clase ***solucion.java***. Utiliza el frame, independiente de la ventana de la aplicación, que se creó en la clase

solucion.java y se encarga de llenarlo de datos para mostrar la información correctamente en la ventana con la solución final.

Cuando un usuario realiza una consulta por cualquiera de las tres ramas de consulta disponibles se llega a esta clase que es llamada siempre desde la clase ***solucion.java***.

Aunque el frame se creó en la clase ***solucion.java***, es esta clase la que se encarga de rellenarla de datos y de su estructura y diseño, excepto en lo que al tamaño se refiere que viene ya definido en la clase ***solucion.java***. Una de las tareas es leer los datos del fichero de configuración de parámetros del programa. Esto se hace realizando una llamada a la clase ***ficheroConfiguracionParametrosSW.java*** para obtener los parámetros por defecto de la aplicación o los parámetros que el usuario guardó en su personalización. Desde este fichero la aplicación será ejecutada en un determinado idioma (inglés o español), con un tamaño de letra y un color en la barra menú.

Esta ventana tiene dos zonas diferenciadas, una de color blanco, que ocupa la mayor parte de la ventana, donde aparecerá la información de la solución y otra zona de color gris, en la parte izquierda y con un tamaño menor, para seleccionar la solución por proceso de COBIT. Además se dispone de una barra de menú especial algo distinta de la barra de la ventana de la aplicación.

La primera vez que aparece esta ventana, en el área blanca, aparecerá un mensaje de texto que dice lo siguiente:

“Choose by clicking the solution from left side.
The solution/solutions are all that have been found in the database through COBIT processes.”

Esto nos indica que pulsando sobre los códigos de procesos (o sobre el proceso, en el caso de que solo exista una solución por un único proceso) que aparecen en la parte gris del lado izquierdo podremos consultar la solución final.

En la parte gris existen tres tooltips para ayudar a entender mejor el funcionamiento de la ventana. Uno de los tooltips sobre los códigos de proceso nos indica que podemos hacer click en una solución mediante el mensaje “Click on a solution”. Hay otro tooltip en la etiqueta “Solution by:” que nos indica que solo se necesita un click mediante el mensaje “Only one click is needed”. El tercer tooltip aparece sobre el signo de interrogación y pone el mensaje “What is process code?” y si se pulsa sobre el icono, el signo de interrogación, aparecerá un panel de aviso para hacer saber a qué proceso corresponde cada código de proceso.

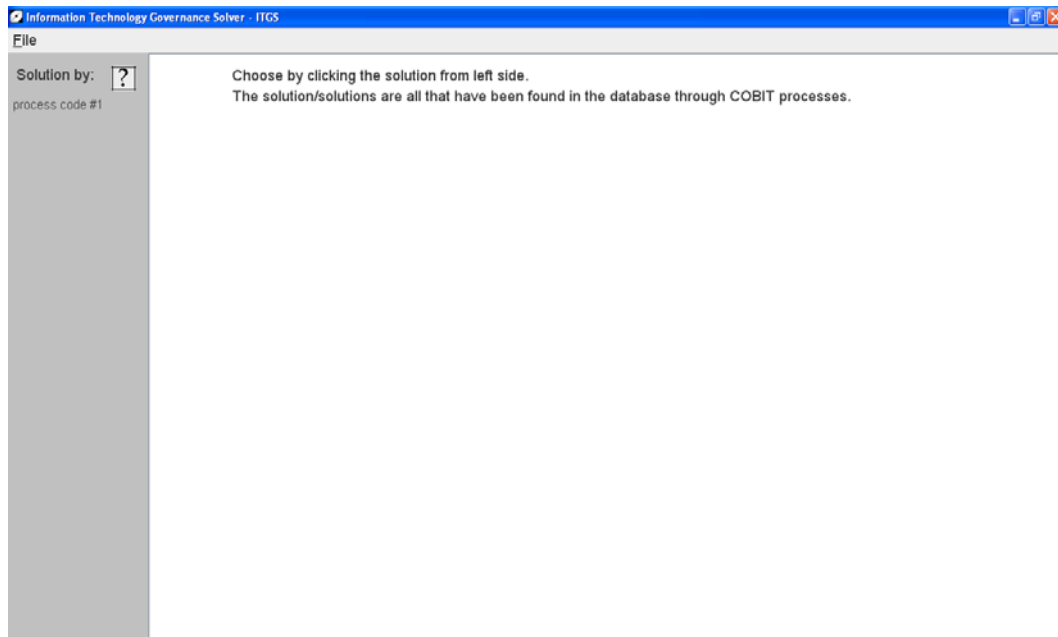


Ilustración 77. Estructura de la ventana ventanaMostrarSolucion

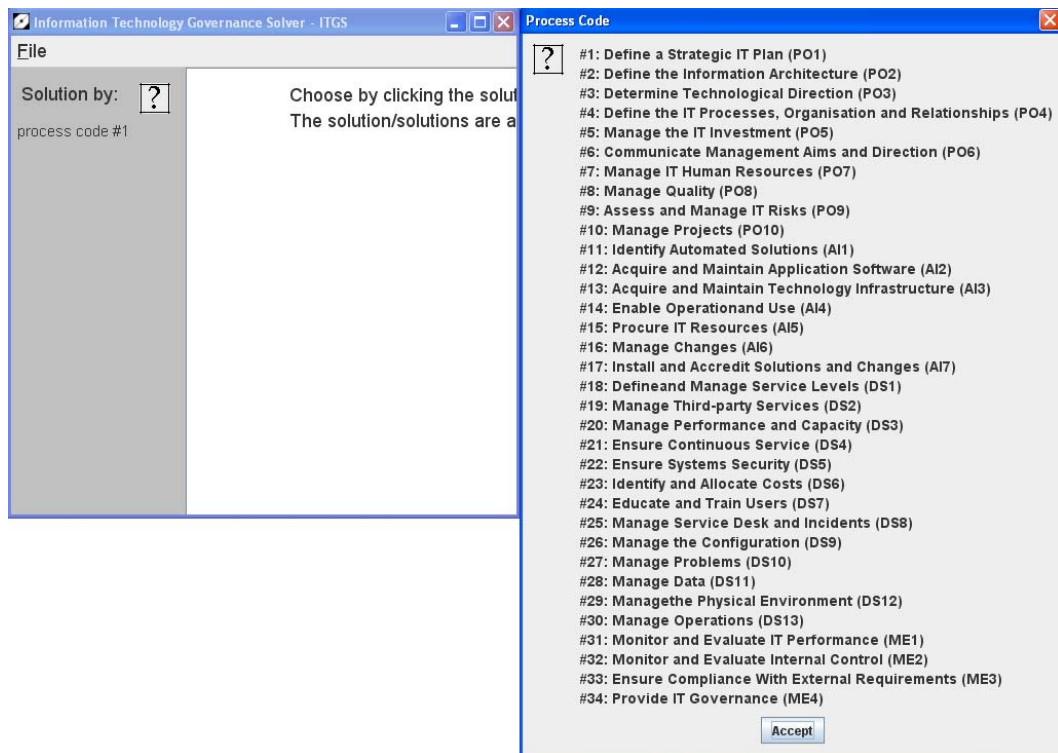


Ilustración 78. Clase VentanaMostrarSolucion con el mensaje de proceso y su código

Cada vez que pulsemos sobre un proceso (existirá una solución) o varios procesos, en el caso de que existan varias soluciones, aparecerá en la parte blanca toda la información que forma parte de la solución y que pertenece a COBIT describiendo las acciones, las metas, las métricas, los objetivos de control, los objetivos TI y los objetivos de negocio necesarios para que la empresa mejore su negocio.

Aunque parezca que la lista de proceso o procesos, que aparece en la parte gris del lado izquierdo de la ventana, pertenece a esta clase por estar usándose aquí, en realidad pertenece a la clase que creó la lista que es la clase ***solucion.java*** pero desde esta clase de la solución (*ventanaMostrarSolucion*) es desde donde se usa por estar visible, es decir que es añadida a la ventana en la clase ***ventanaMostrarSolucion.java*** pero no es aquí donde fue creada, y sin embargo es aquí donde tiene asociado el evento que responde a las pulsaciones sobre la lista.

Cada vez que pulsemos sobre un proceso o código de proceso se produce un evento y se realiza una llamada a la clase ***solucion.java*** y concretamente al método *obtenerSolucionProceso* al cual le enviamos el proceso seleccionado para que nos devuelva toda la información de COBIT y muestre el texto en la parte blanca correctamente tratado. Esto se repetirá tantas veces como se seleccione un proceso.

La razón por la cual se usan variables entre dos clases, que en principio son muy similares ya que intervienen en la solución y se necesitan la una a la otra para funcionar correctamente, es porque algunos elementos se necesitan que sean creados una sola vez, y otros elementos sean creados y actualizados respondiendo a eventos en una ventana. Algunos de estos elementos, como la lista de procesos, si estuvieran en la misma clase de la ***ventanaMostrarSolucion.java*** se perderían porque esta ventana necesita un refresco o actualización para mostrar la parte blanca con la información que cambia cuando se selecciona un código de proceso. Es por ello que la lista de procesos es mejor que esté en otra clase independiente, pero sin embargo se requiere poder usarla desde esta clase y es lo que se permite.

Los objetivos TI y los objetivos de negocio aparecen sin información en la ventana de la solución debido a que no existe en el documento COBIT información explícita acerca de los objetivos TI. Sí existe información en COBIT acerca de los objetivos de negocio (cuya información está introducida en la base de datos) que a su vez aparecen vacíos en la solución y la razón es porque los objetivos de negocio están relacionados con los objetivos TI, y al no tener datos sobre los objetivos TI, con el actual diseño de la base de datos, los objetivos de negocio no tienen ninguna relación establecida y es por ello que no aparecen tampoco en la solución. Sin embargo se ha optado por dejar ambos campos (objetivo TI y objetivo de negocio) y permitir la introducción de datos sobre ambos y sus relaciones para que en un futuro pueda existir esta información.

La ventana **ventanaMostrarSolucion.java** dispone de barra de desplazamiento vertical y horizontal para que en el caso de que la información sobresalga del tamaño de la ventana pueda leerse enteramente. Se ha hecho un tratamiento de los caracteres para que se pueda visualizar la mayor parte posible de información sin tener que hacer uso de la barra de desplazamiento horizontal, sin embargo en algunas ocasiones habrá que utilizarla para poder leer todo el texto completo. Aparte se tiene en cuenta tratar el texto con un interlineado que facilite la lectura de cada línea.

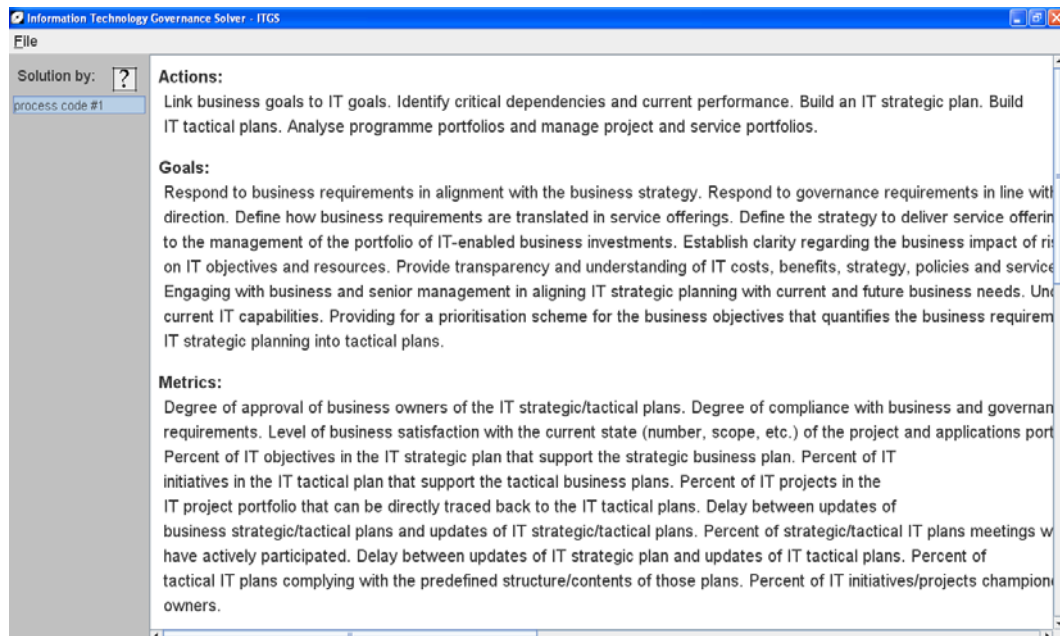


Ilustración 79. Ventana ventanaMostrarSolucion con la solución de COBIT

En cuanto a la barra de menú de esta ventana, es algo distinta a la barra de menú de la ventana de la aplicación. La barra de menú en este caso solo tiene el menú File que está formado por tres ítems. El ítem “Save as...” que sirve para guardar o exportar la solución a un fichero físico, el ítem “Print” que se utiliza para imprimir y el ítem “Exit” para salir de la ventana. Los ítems “Save as...” y “Print” aparecen habilitados en esta ventana de la solución, en la ventana de la aplicación aparecían inhabilitados ya que es en la ventana de la solución donde tienen sentido usarlos. Concretamente aparecen habilitados cuando se pulsa una solución (código de proceso) en la parte izquierda de color gris, hasta entonces aparecerán, pero también inhabilitados.

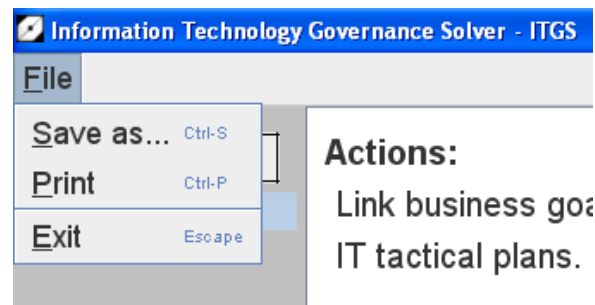


Ilustración 80. Menú File en la ventana ventanaMostrarSolucion

La opción de imprimir solo imprime una hoja y se ha dejado el ítem a modo de demostración. Lo que hace esta clase interna es realizar un tratamiento de la información que tiene la clase **ventanaMostrarSolucion.java** que a su vez posee esta información obtenida desde la clase **solucion.java**. Esta información trata de las acciones, metas, métricas, objetivos de control, objetivos TI y objetivos de negocio, y es tratada para que el texto salga en una hoja de modo que con saltos de línea pueda leerse correctamente sin que sobresalga el límite de una hoja de papel. Sin embargo por problemas en el código en esta parte, que no es algo simple, no se ha conseguido por falta de tiempo realizar una impresión del resto de páginas. Pero se permite guardar el archivo desde el ítem “Save as...” y desde ahí se puede imprimir sin problemas.

La opción de exportación, salvado o guardado en fichero está disponible mediante el ítem “Save as...”. La solución mostrada en esta ventana puede salvarse en tres tipos de ficheros, uno con extensión TXT, otro con extensión PDF y otro con extensión HTML. Se han elegido estos tipos de fichero por su popularidad y por el hecho de que pueden ejecutarse mediante software libre y gratuito.

Cuando pulsemos para guardar la información aparecerá una ventana donde podemos elegir un tipo de extensión, un nombre para el fichero, una estructura de ventanas para elegir la ruta donde se quiera guardar el archivo, y un botón para salvar “Save” o para cancelar “Cancel”. Es una ventana típica de exportación de ficheros que el propio Java proporciona (JFileChooser) y se encarga de realizar la estructura de la ruta de los ficheros y carpetas de un ordenador, los ficheros que aparecerán serán los que dependiendo de la extensión seleccionada se encuentren en una ruta dada. Por defecto la primera extensión que aparece es la de TXT.

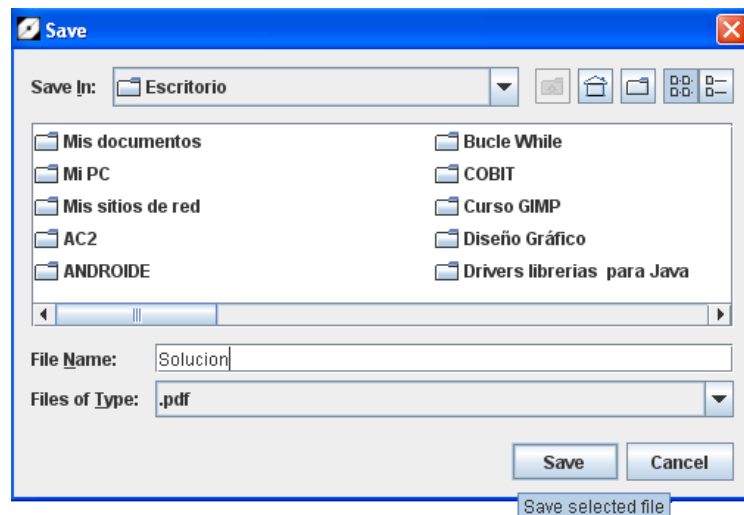


Ilustración 81. Salvar en fichero

El nombre se puede introducir en el área habilitado para ello o bien seleccionar un nombre de un fichero existente en una ruta determinada.

Para cada tipo de fichero que se seleccione se comprueba si ya existía en la ruta que se desee guardar, en caso negativo creará un nuevo fichero y en caso de que ya existiera, se avisará mediante un panel de que el archivo ya existe y se da la opción de sobrescribirlo o no. Si se decide sobrescribir un fichero existente, la aplicación borrará el fichero antiguo y creará uno nuevo con la nueva información de la misma forma que si el fichero no existiera.

Si el fichero no existe y se tiene que crear de nuevo, el proceso de guardado es similar para cada tipo de fichero pero con algunas diferencias en el modo de escribir la información en el fichero.

El proceso para cada tipo de fichero es el siguiente. Se realiza un tratamiento de la información que tiene la clase **ventanaMostrarSolucion.java** que a su vez posee esta información obtenida desde la clase **solucion.java**. Esta información trata de las acciones, metas, métricas, objetivos de control, objetivos TI y objetivos de negocio, y es tratada mediante saltos de línea para que el texto aparezca correctamente en el fichero y no sea necesario el uso de una barra de desplazamiento horizontal. Se cuentan las palabras por cada frase y al llegar a un tope en el que la frase se podría salir del límite de la pantalla se inserta un salto de línea.

La diferencia entre cada tipo de fichero es que en el caso de guardar un fichero en formato TXT, se puede usar una instrucción típica de escribir sobre fichero en Java.

En cambio si se guarda en formato PDF esta instrucción típica no es válida y ha de usarse una librería para el tratamiento de ficheros PDF. La librería usada en la aplicación es una librería con licencia gratuita llamada iText-5.0.6 de iText. Gracias a

esta librería se pueden usar instrucciones en Java que permiten manipular o crear archivos escritos en formato PDF.

En el caso de guardar la información en formato HTML se crea una estructura típica de este formato que es:

```
<html>
<head>
<title>
    Aquí se introduce el nombre de la aplicación.
</title>
</head>
<body>
    Aquí se escribe toda la información de la solución.
</body>
</html>
```

Además la instrucción para guardar información en un fichero HTML también es distinta al tipo de fichero TXT y PDF.

A continuación se muestra como se visualizaría la solución en estos tres tipos de ficheros:

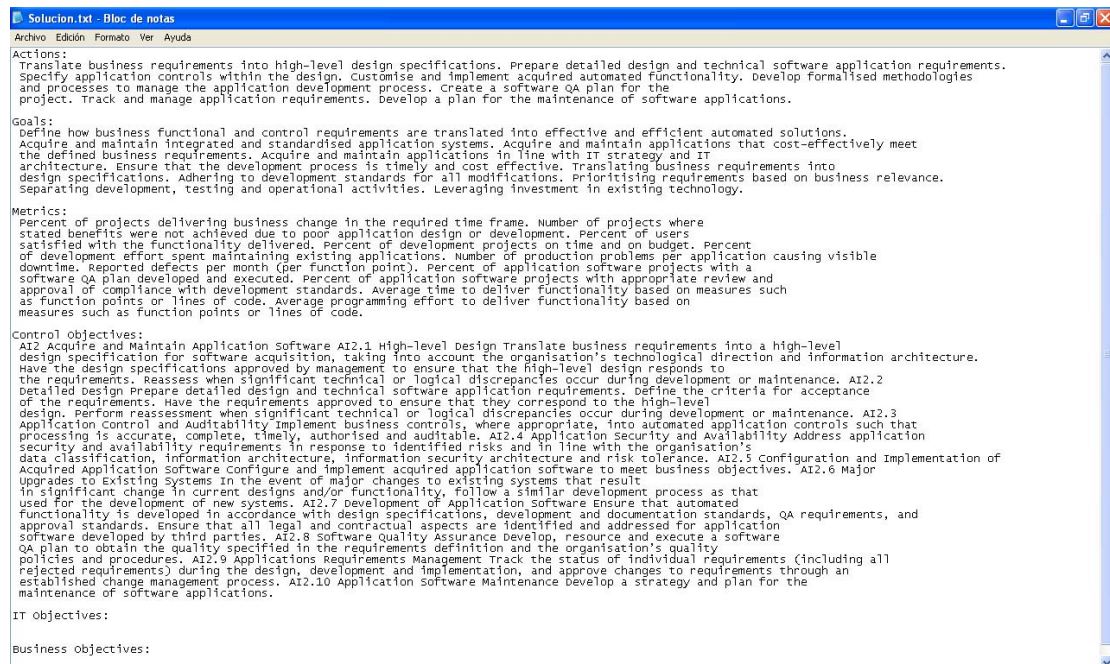


Ilustración 82. Solución en fichero TXT

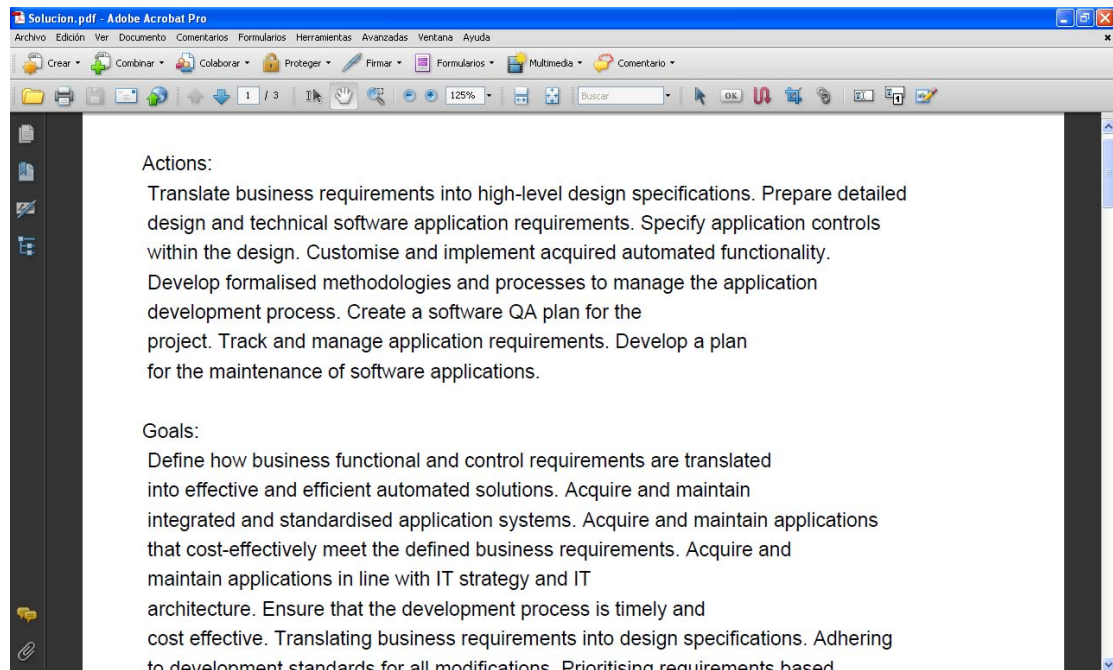


Ilustración 83. Solución en fichero PDF

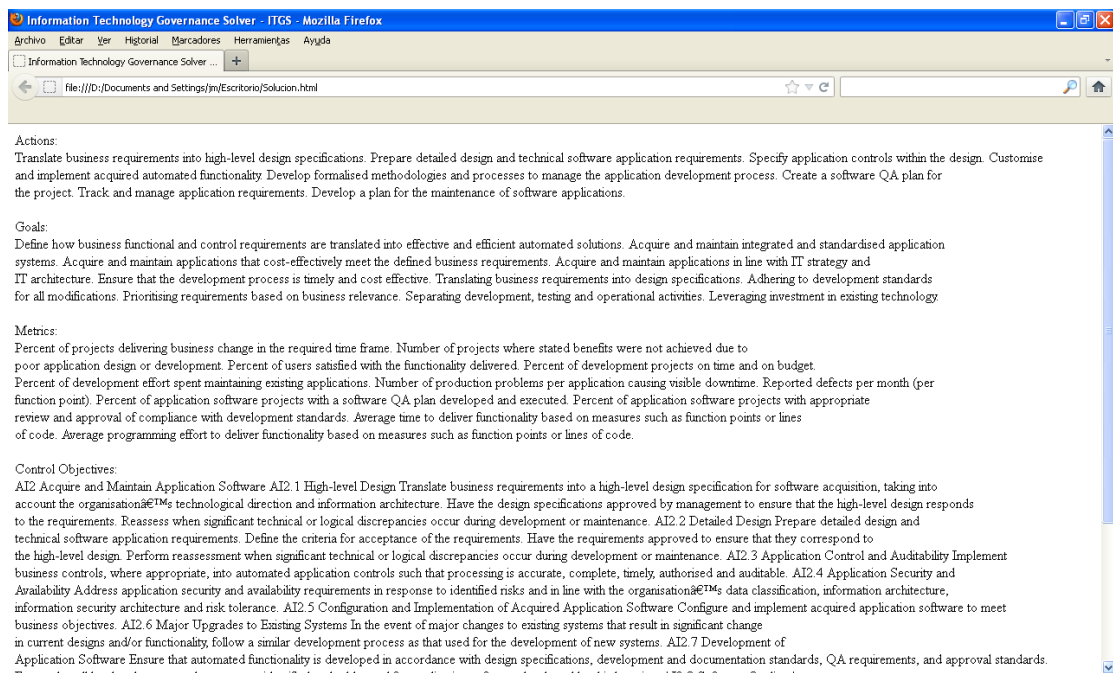


Ilustración 84. Solución en fichero HTML

Clase ventanaAdministrador.java:

La ventana administrador es la ventana para poder realizar la administración del programa, en general relacionadas con el uso de la base de datos con la que la aplicación se comunica. El administrador se deberá loguear o hacer login, es decir introducir su contraseña identificativa de administrador, en el sistema de forma que pueda tener acceso a realizar modificaciones en la base de datos o en la aplicación propias de un perfil de administrador, que un usuario normal, al que solo se le tienen permitido tareas de consulta y configuración de preferencias, no puede realizar. Es por ello que el administrador debe tener una contraseña de seguridad (por defecto esta contraseña es 'password') ya que cualquier usuario que tuviera acceso a esta parte podría dañar el sistema voluntaria o involuntariamente, puesto que una mala configuración de la administración podría afectar en el correcto funcionamiento del programa. Si no se desea acceder al perfil administrador se dispone del botón "Back", para volver hacia la ventana anterior que sería la ventana principal, o del menú "Menu" para acceder a cualquier otra rama del programa.

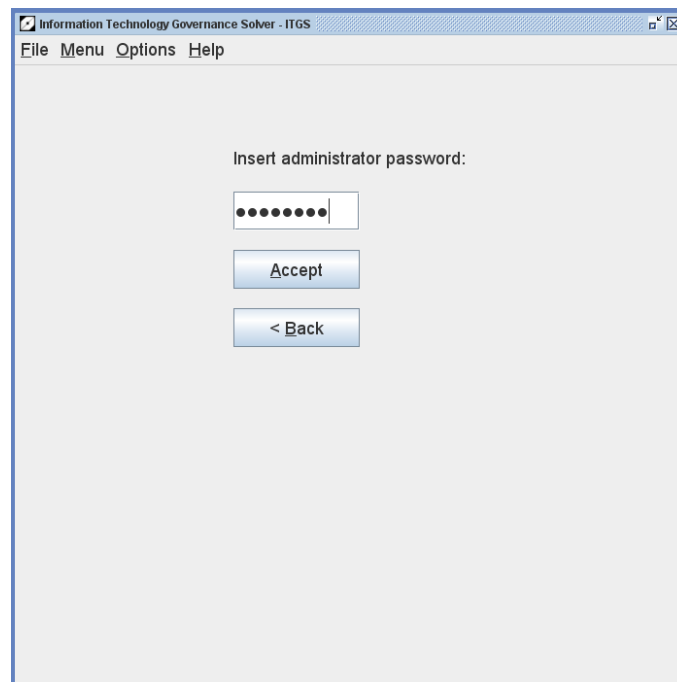


Ilustración 85. Acceso por contraseña del perfil administrador en la ventana "Administrador"

Cuando el administrador introduce su contraseña, protegida mediante caracteres especiales, y tras pulsar el botón “Accept” (o presionando la tecla Enter) es comparada con la contraseña guardada previamente en un fichero cifrado (ficheroPasswordAdministrador.txt) y si coinciden entonces el administrador es validado correctamente. En este caso, el sistema muestra una serie de opciones que se pueden configurar, todas ellas relacionadas con la base de datos y también con la propia contraseña del administrador. Esta ventana se crea mediante la clase **ventanaAdministrar.java** cuando un administrador ha sido validado correctamente. Desde esta ventana (ventana administrador) es posible también volver al menú principal mediante el botón “Main menu” o mediante el uso de la barra de menú.

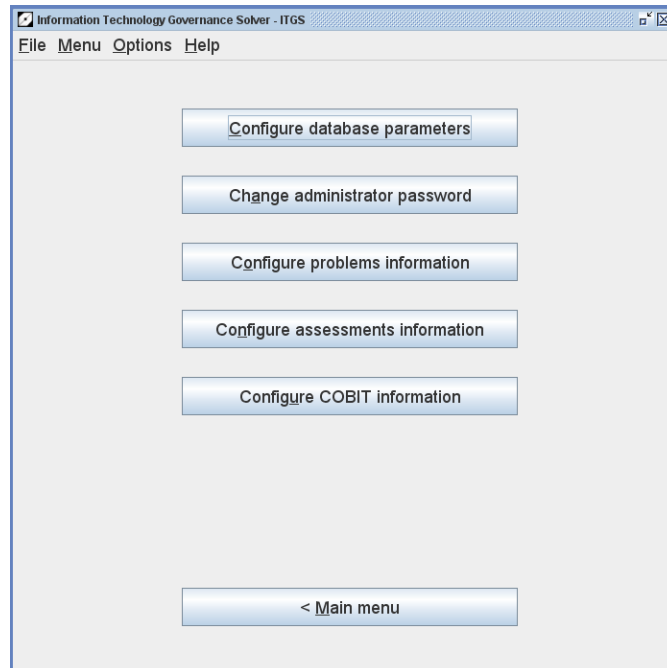


Ilustración 86. Tareas del administrador

Clase ventanaAdministrar.java:

Esta ventana permite la modificación de parámetros en la base de datos y de la contraseña del administrador. A cada una de las tareas posibles se accede mediante botones que al ser pulsados generarán nuevas clases, esto es, nuevas ventanas, que irán permitiendo realizar todas las gestiones disponibles. Las tareas que el administrador puede realizar sobre el sistema son:

- Configurar los parámetros de la base de datos.
- Cambiar la contraseña del administrador.
- Configurar la información de los problemas.
- Configurar la información de la evaluación.
- Configurar la información de COBIT.

Configuración de los parámetros de la base de datos:

Cuando pulsamos el botón “Configure Database parameters” se crea la clase **ventanaConfigurarBaseDatos.java**. Desde esta nueva ventana se pueden cambiar los valores con los que la aplicación se comunica con la base de datos. Los campos que se necesitan para configurar una conexión con una base de datos son Host, Port, Database, User Name y Password, esto es lo que se conoce como Database URL y consta de la dirección IP del ordenador (anfitrión), el puerto de comunicación con la base de datos (normalmente el puerto de red 1521), el nombre de la base de datos (en nuestro caso “orcl”, de Oracle), el nombre del usuario de la base de datos (system) y la contraseña de la base de datos (pass). Para ayudar al administrador se pueden ver los campos con los valores actuales de la configuración de la conexión que están guardados previamente en un fichero cifrado (*ficheroConfiguracionBaseDatos.txt*) cuando se accede a esta ventana de configuración. En el caso de que se introduzcan valores vacíos o el carácter ‘espacio’ el sistema avisará indicando que esto no es posible y que debemos introducir valores correctos. Cada caja de texto que permite la modificación e introducción de datos tiene un límite de 20 caracteres.

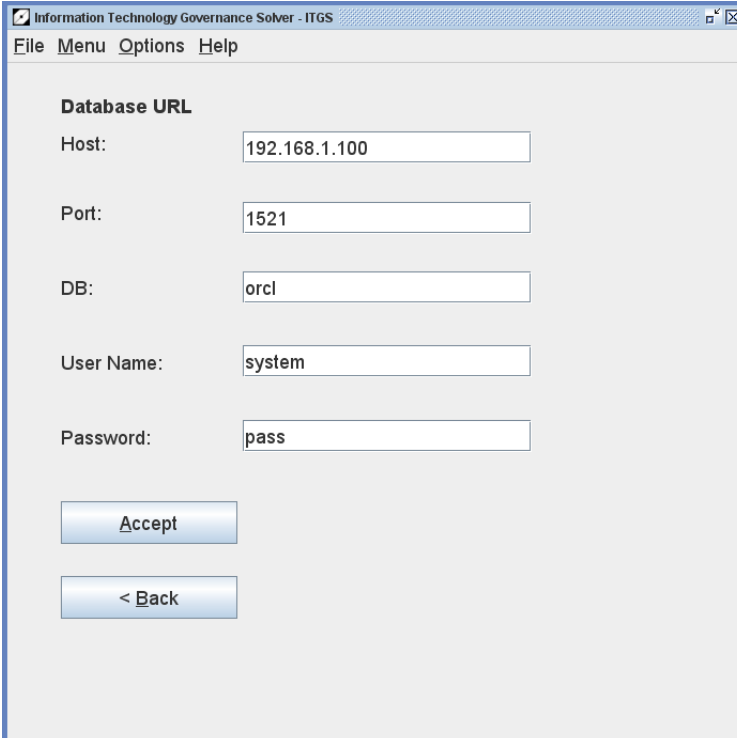


Ilustración 87. Parámetros configurables de la base de datos

Cuando se ejecuta la aplicación, una de las primeras cosas que el programa comprueba es que la conexión con la base de datos sea correcta, ya que si no existe una conexión con la base de datos el sistema no funcionará bien. Por tanto, si el sistema comprueba que la conexión ha fallado nos lo indicará con un mensaje de aviso.

En el caso de que no haya una conexión con la base de datos e intentemos usar el programa en modo de perfil de usuario normal, el programa nos avisará del problema, nos dirá que contactemos con el administrador y a continuación se cerrará como medida de seguridad, tras pulsar en el botón Accept o hagamos click en el recuadro X (cerrar) de la ventana de aviso.



Ilustración 88. Mensaje de error de conexión con la base de datos usando el programa con el perfil usuario

Si intentamos usar el programa en el modo administrador, cuando sigue existiendo un problema en la conexión de la base de datos, se nos pedirá la contraseña y la aplicación nos llevará directamente hasta la ventana de configuración de la base de datos, sin dejarnos ver más opciones de administración, para permitirnos verificar y comprobar los datos y establecer una conexión correcta. En este caso se creará la clase ***ventanaConfigurarBaseDatosError.java (VCBDE)***, que es muy similar a la clase y ventana ***ventanaConfigurarBaseDatos.java (VCBD)*** y que se crea cuando existe un error en la conexión, cuando no hay error la ventana que se crea es la VCBD. La principal diferencia es que la ventana VCBDE muestra en la parte inferior el botón "Main menu" para volver al menú principal y en cambio la ventana VCBD muestra en la parte inferior el botón "Back" en lugar del botón "Main menu", ya que si existe error no se permite cambiar otros datos que no sean los de la conexión, por lo que solo se permite ir hacia el menú principal. Pero cuando la conexión es correcta se permiten seguir cambiando datos por lo que el botón "Back" se habilita para poder ir hacia atrás

en cuyo caso se visualizaría la ventana que permite cambiar todos los datos que es la ventana **ventanaAdministrar.java**. Cuando cambiemos los datos de la conexión con la base de datos el sistema volverá a comprobar la validez de la conexión y nos indicará si esta vez es correcto con lo que podremos utilizar la aplicación de un modo normal o en caso contrario volverá a aparecer la ventana de configuración para que sigamos cambiando los datos de conexión ya que siguen siendo incorrectos. En todo caso se alternan la VCBDE con la VCBDE en el caso de que la conexión sea correcta o incorrecta respectivamente, en cada comprobación de validez en la conexión.

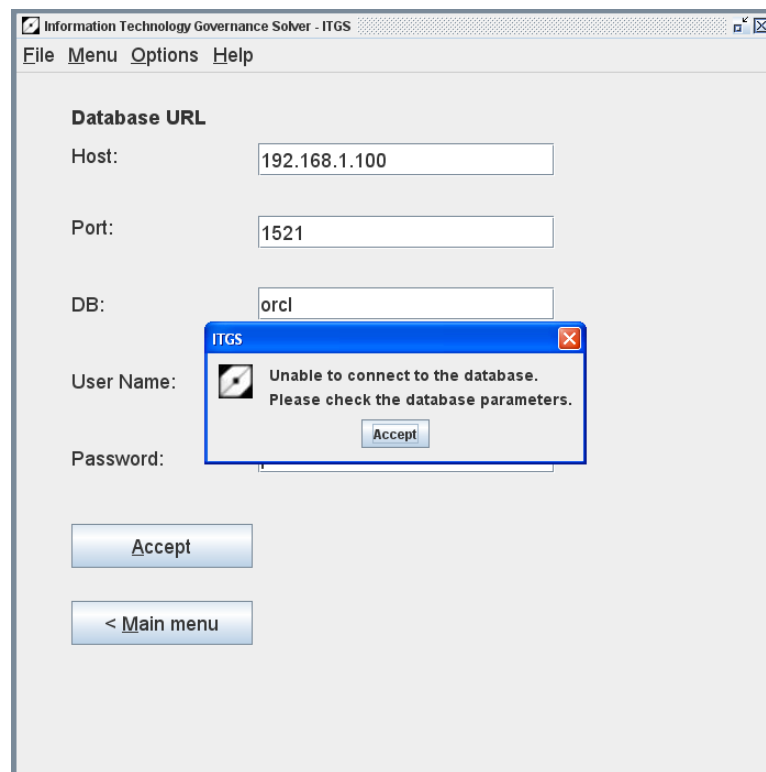


Ilustración 89. Mensaje de aviso de conexión incorrecta con la base de datos usando el perfil administrador

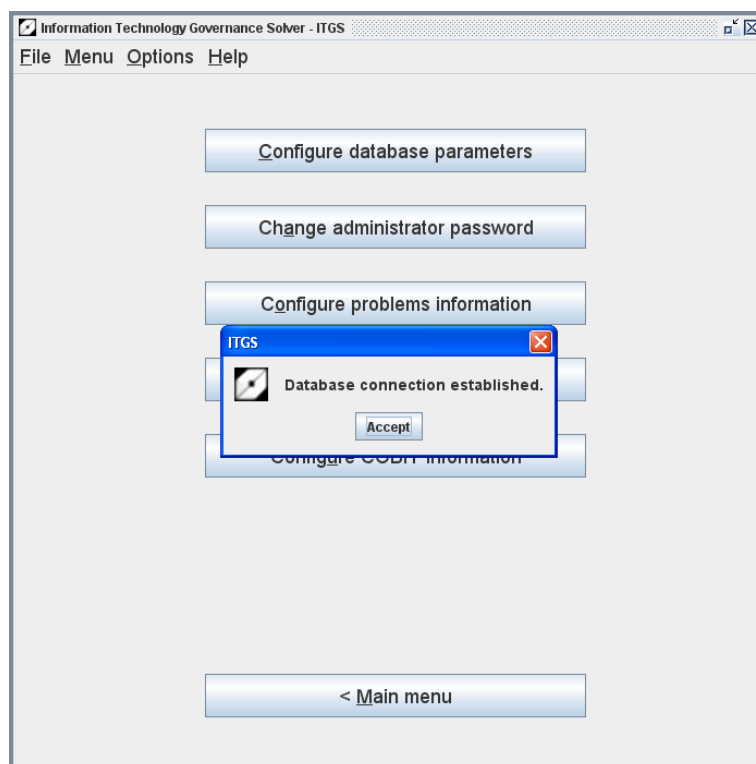


Ilustración 90. Mensaje de aviso de conexión correcta establecida con la base de datos

Si se desean cambiar los datos de la conexión, cuando el programa se conectó correctamente el procedimiento en todos los casos es el mismo. Acudiríamos a esta ventana y una vez introducidos los datos pulsaríamos el botón “Accept” (o pulsamos la tecla Enter). A continuación la aplicación modificaría el fichero cifrado que contiene los datos de la conexión (ficheroConfiguracionBaseDatos.txt) e intentaría conectarse haciendo una llamada a la clase **conexion.java**, y a partir de ahí si la conexión es correcta o no se repetiría el proceso comentado anteriormente.

Cambiar la contraseña del administrador:

Cuando pulsamos el botón “Change administrator password” se crea la clase **ventanaCambiarContraseñaAdministrador.java**. Desde esta clase se crea la ventana que permite el cambio de la contraseña del administrador. Se pide en primer lugar insertar la antigua contraseña y a continuación la nueva contraseña, más una vez más la nueva contraseña para verificar la coincidencia entre ambas y asegurar que la nueva contraseña es la que realmente el administrador quiere utilizar. Las cajas de texto son de tipo `JPasswordField` para proteger mediante caracteres especiales los caracteres reales que son introducidos como medida de seguridad. Por otra parte, se controla que se introduzca la contraseña antigua y sea la correcta, así como que se introduzcan valores (valores vacíos no son permitidos) para la contraseña nueva y su confirmación y no contengan el carácter ‘espacio’, y que la nueva y la confirmación de la nueva

contraseña coincidan. El límite de texto para las cajas de texto es de 20 caracteres. Cuando se pulsa el botón “Accept” (o pulsamos la tecla Enter), el sistema modificará el fichero cifrado (ficheroPasswordAdministrador.txt) para guardar la nueva contraseña. En este caso, al contrario que en las cajas de texto de la ventana de cambiar la configuración de la conexión de la base de datos, no se realiza una lectura previa de fichero para mostrarla por pantalla ya que no tiene sentido mostrar la contraseña del fichero si se quiere tener mayor seguridad.

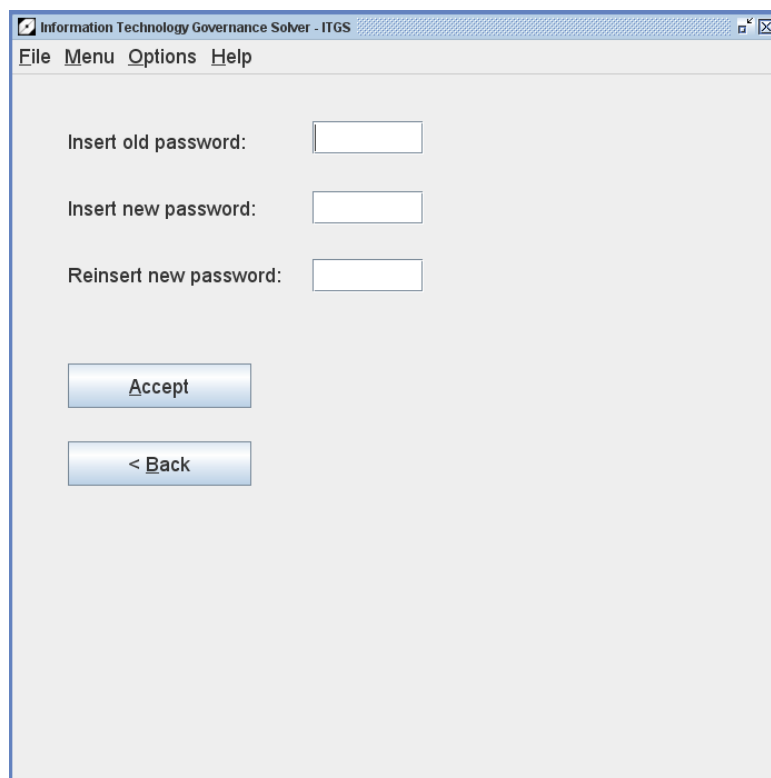
The image shows a screenshot of a software application window titled "Information Technology Governance Solver - ITGS". The window has a menu bar with "File", "Menu", "Options", and "Help". The main area contains three text input fields with labels: "Insert old password:", "Insert new password:", and "Reinsert new password:". Below these fields are two buttons: "Accept" and "< Back".

Ilustración 91. Cambio de la contraseña del administrador

Configurar la información de los problemas, la información de la evaluación, y la información de COBIT:

Estas tres tareas disponibles en la administración contienen las mismas opciones de configuración de base de datos que son el alta (Add information), la baja (Delete information) y la modificación de información (Update information). Dentro de la modificación hay dos opciones que son el cambio de nombre o descripción (Change names) de la información y la creación o borrado de relaciones entre tablas (Relate database problems/assessments/Cobit tables). Lo que cambia principalmente, con algunos pequeños detalles distintos de tratamiento que se explicarán más adelante, es que en cada una de las tres tareas la información es distinta, una trata de la información de los problemas, otra de la información de las evaluaciones (assessments) y la otra trata de la información de COBIT 4.1. Esta información es la que

se usa en cada una de las tres ramas de consulta de que dispone el usuario así como de la información que se obtiene en la solución de cada una de las tres ramas de consulta.

Se permite de esta manera añadir información en el alta, borrar información en la baja y modificar la información en la modificación (*update*). Es importante señalar que se permite introducir nuevos datos sin estar relacionados, es decir que se puede por ejemplo introducir un problema y unas palabras para ese problema y sin embargo no estarán relacionados hasta que en la modificación creamos esa relación entre tablas de la base de datos. De esta manera la administración permite introducir datos sin ser restrictiva y deja al administrador la tarea de relacionar posteriormente la información si fuera necesario. Es importante relacionar las tablas de las tuplas nuevas de información que se hayan dado de alta ya que si no está relacionada la información nueva, repercutirá no apareciendo en la solución ni en las ramas de consulta, o apareciendo pero no funcionando como se deseaba, aparecerán avisos indicando este problema, ya que están a la espera de ser relacionadas para funcionar correctamente.

En las altas además se comprueba que la información no sea repetida avisando con un mensaje en caso positivo o negativo y hay que tener en cuenta que se diferencia entre minúsculas y mayúsculas, por lo tanto no es lo mismo “Plan and Organise” que “plan and organise”. Por otra parte siguiendo en las altas, se controla mediante un limitador de caracteres el número de caracteres que se permiten introducir como máximo y se controla además que la información introducida sea correcta, es decir que no sea vacía y que esté libre de caracteres especiales, mediante un tratamiento de los caracteres, que puedan dar problemas de lo que se conoce como “inyección SQL”. Existe un apartado en el que se habla sobre la problemática de la inyección SQL y su solución.

También hay que decir que cuando se produce un borrado de información (una tupla en la base de datos) automáticamente las relaciones que tuviera esa tupla con otras tuplas se borrarán ya que la base de datos está configurada en el modo borrado en cascada (On delete cascade). En el borrado se ha implementado una casilla de “Select all” para facilitar la tarea de seleccionar todas las tuplas a la vez o si se pulsa de nuevo para deseleccionar todo a la vez, o si se quiere, seleccionar o deseleccionar una a una también es posible.

En la administración de la modificación, en concreto en la parte de modificación de cambio de nombres, se usan unos menús desplegables desde los cuales se puede seleccionar la tupla deseada apareciendo en el lado izquierdo en una caja de texto para que se pueda modificar con mayor comodidad, tanto si se trata de una modificación de pocos caracteres o de todos. En la parte de la modificación referente a la relación entre tablas, el programa permite elegir mediante menús desplegables las tablas requeridas y comprueba si ya existían o no relaciones entre ellas avisando con mensajes en pantalla.

Los números que aparecen entre corchetes al lado de la información en los menús desplegables tanto en cambio de nombres como en relación de tablas son códigos de tratamiento interno para la base de datos que no tienen ninguna relevancia pero que sirven para control interno de la base de datos y en caso de sobrepasar un cierto número avisará con un mensaje de que se ha sobrepasado el tamaño máximo de la base de datos y que es necesario contactar con un administrador o borrar tuplas para recuperar espacio. Estos números no tienen por qué ser consecutivos ya que dependen de cómo se realicen las tareas de inserción y borrado por lo que no son motivo de interés. Como curiosidad este número es considerablemente alto y habría que sobrepasar más de 999.999.999 tuplas para empezar a tener problemas de espacio en la base de datos por número de tuplas utilizadas. Los códigos numéricos empleados en la base de datos son de tipo *NUMBER*, tipo usado para introducir valores numéricos, de 9 dígitos, por lo que el mayor valor que soportarían es el indicado anteriormente. Únicamente en los menús que representen la información de las palabras y los sinónimos en la administración de los problemas no presentarán estos números ya que esta información en la base de datos no utiliza un código numérico, sino que el código es la propia palabra o sinónimo que es único y son caracteres.

A continuación se procede a explicar con mayor detalle cada clase de las comentadas de la administración.

Configuración de la información de los problemas:

Cuando pulsamos el botón “Configure problems information” desde la “ventana Administrar” se crea la clase ***ventanaAdministrarProblema.java*** que permite las tareas de dar de alta, de baja y de modificación mediante los botones “Add information”, “Delete information” y “Update information” respectivamente. También se permite volver hacia atrás mediante el botón “Back” con el que se regresaría a la “ventana Administrar” y se permite volver al menú principal mediante el botón “Main menu”.

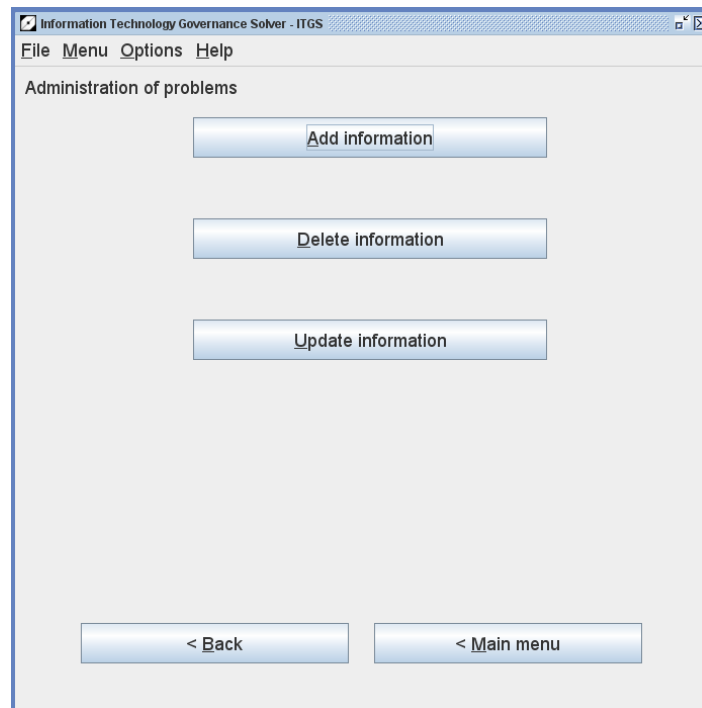


Ilustración 92. Altas, bajas y modificación (administración) de información de los problemas

Cuando pulsemos cualquiera de los botones de las altas, bajas o modificaciones se crearán las siguientes clases para cada una de ellas. El nombre dado a las clases de la administración en la parte de información de los problemas ha sido:

- Para las altas:

ventanaAdministrarProblemaAlta.java

- Para las bajas:

ventanaAdministrarProblemaBaja.java

- Para las modificaciones:

ventanaAdministrarProblemaModificacion.java

- Para el cambio de nombres:

ventanaAdministrarProblemaModificacionNombre.java

- Para la relación entre tablas:

ventanaAdministrarProblemaModificacionRelacion.java

En esta ventana se puede administrar todo lo relacionado con los problemas de la base de datos. Los problemas son los problemas TI, que a su vez tienen una serie de palabras y sinónimos relacionados.

La clase ***ventanaAdministrarProblemaAlta.java*** permite realizar las altas sobre los problemas TI de la base de datos, así como de las palabras y sinónimos que pueda contener dicho problema. La ventana es creada al pulsar sobre el botón “Add information” desde la anterior clase ***ventanaAdministrarProblema.java***. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana “administrar Problema”) y otro (“Main menu”) para ir al menú principal. Después se encuentran una serie de cajas de texto para añadir toda la información que puede existir en la base de datos sobre los problemas, palabras y sinónimos del problema. La ventana dispone de una barra de desplazamiento vertical (scroll vertical) para poder visualizar toda la ventana entera sin problemas.

Para dar de alta a un problema, se dispone de un área de texto con un límite de 4000 caracteres y con un scroll vertical en caso de que la información sea extensa para poder visualizarla completamente.

Después se tiene una caja de texto, con un límite de 2000 caracteres, para introducir palabras que pueda contener un problema.

Y por último se tiene una caja de texto, con un límite de 2000 caracteres, para introducir sinónimos que pueda tener una palabra.

La razón de que palabra y sinónimo no tengan un límite también de 4000 caracteres cada una es que la tabla intermedia que comparten ambas tablas de palabra y sinónimo contiene los códigos primarios de ambas tuplas que son los 2000 caracteres cada una y para que esto no traiga problemas de límite con la base de datos, si sumamos las dos se tiene un total de 4000 caracteres para la tabla intermedia lo cual es correcto. Pero si las tablas palabra y sinónimo tuvieran 4000 caracteres, entonces la tabla intermedia tendría un total de 8000 caracteres entre las dos y esto supera el límite en la base de datos y daría lugar a problemas si introdujésemos 4000 caracteres para palabra y sinónimo, ya que en la tabla intermedia se estarían introduciendo 8000 caracteres, lo cual no es válido para instrucciones típicas de inserción, habría que hacer inserciones mediante un procedimiento especial en la base de datos.

Tanto las palabras como los sinónimos pueden ser introducidos varios a la vez, no es necesario introducir palabra por palabra ni sinónimo por sinónimo, siempre que los separemos mediante al menos un espacio. De esta manera se facilita la tarea al administrador en caso de tener que introducir muchos datos en esas partes.

Como se ha dicho anteriormente, el alta de la información no es restrictiva y permite introducir datos para posteriormente ser relacionados. Por ello, cuando introducimos un problema o unas palabras o sinónimos, no es necesario que estén relacionados entre sí, ya que las relaciones se deben hacer después en la modificación de tablas.

Debajo de cada caja de texto existe un botón de aceptar “Accept” para enviar la nueva información a la base de datos. Cuando se pulsa en el botón correspondiente se realizan una serie de acciones que son las de tratamiento de la cadena introducida por el administrador y una vez tratada se hace una inserción en la base de datos siempre que la información no sea repetida en la base de datos. Si se pulsa sin información introducida en la caja de texto el sistema nos indicará con un mensaje de que introduzcamos información válida. Cuando existe información, el tratamiento de la cadena se ocupa de quitar los espacios que no aportan información del comienzo y del final, y elimina todos los signos que pudieran provocar una inyección SQL, permitiendo solo el uso de algunos caracteres especiales, por lo que si solo se introducen caracteres no permitidos el sistema nos avisará de que se introduzcan valores válidos. También se comprueba que no sea una cadena vacía y todo ello mediante mensajes de aviso. En el caso de introducir un problema, cuando la cadena de caracteres está libre de posibles caracteres problemáticos se realiza un aumento en el código de la tupla de la base de datos, teniendo en cuenta el valor máximo de los códigos ya existentes y añadiéndole un número más de incremento. En el caso de que no hubiera tuplas, no existiría valor máximo y simplemente en caso de ser la primera tupla insertada se iniciaría el código con el valor 1. Antes justo de la inserción, se comprueba que este número incrementado no supere el valor máximo existente en la base de datos, en cuyo caso se avisaría, y si el número no se supera, se comprueba que la información nueva no se encuentre ya en la base de datos. Si está ya en la base de datos no se insertará y se avisará con un mensaje y en caso de que no esté en la base de datos se realiza la inserción del problema.

Para la información de palabra y sinónimo se realiza el mismo tratamiento de la cadena de caracteres pero no se incrementa ningún valor de su código como en problema, ya que el código de palabra y sinónimo no es numérico, sino que son los caracteres de la propia palabra y sinónimo, que sirven tanto de código como de descripción puesto que es único. Cada palabra y cada sinónimo son únicos.

En la inserción de palabra y sinónimo, puesto que se permite introducir varias tuplas al mismo tiempo tiene un algoritmo distinto. La cadena introducida es tratada como una lista de palabras o sinónimos, dependiendo del caso (realmente no es una lista sino que es un tipo *String*, cadena de caracteres, pero el tratamiento mediante *tokens* de Java permite usarlas como si de un tipo *lista* se tratara). Se comprueba cada palabra (o sinónimo) de la lista con cada palabra (o sinónimo) de las existentes en la base de datos y se divide en dos listas, una lista con las palabras (o sinónimos) encontradas y que no serán insertadas y otra lista con las palabras (o sinónimos) no encontradas que serán las que se inserten. En el caso de que existan palabras (o

sinónimos) en la base de datos se avisará por mensaje de qué palabras no van a ser insertadas y en el caso de que haya palabras (o sinónimos) nuevas se introducirán en la base de datos.

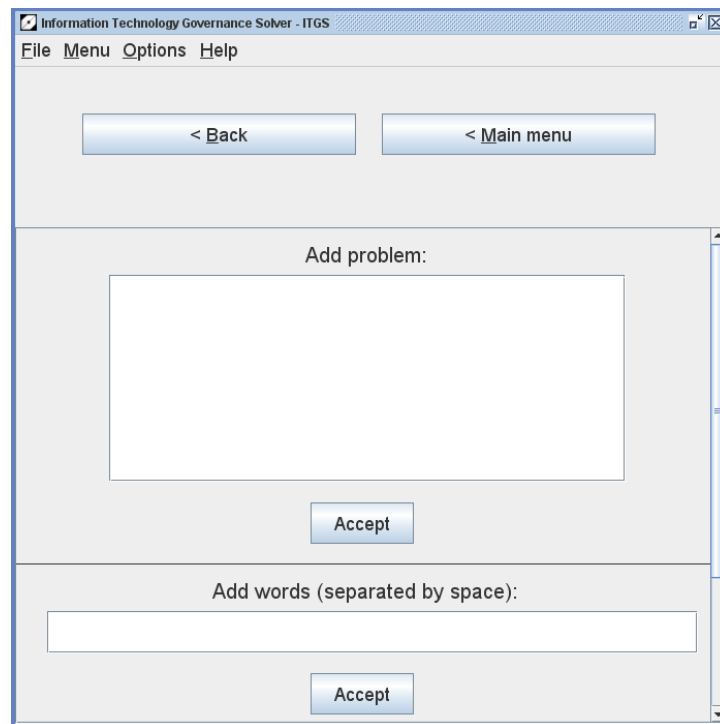


Ilustración 93. Alta en administración de problemas

La clase **ventanaAdministrarProblemaBaja.java** permite realizar las bajas en la información de los problemas TI, palabras y sinónimos relacionadas y la ventana es creada al pulsar sobre el botón “Delete information”. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar Problemas) y otro (“Main menu”) para ir al menú principal. Después se encuentran una serie de cuadrados seleccionables o checkbox que permiten la selección de varios elementos a la vez. La ventana posee una barra de desplazamiento vertical y horizontal para visualizar todo el contenido correctamente. En esta ventana aparece toda la información existente de problemas, palabras y sinónimos. Cuando se crea la ventana toda esta información se obtiene desde la base de datos y se representa por pantalla.

Se pueden seleccionar o deseleccionar cada una de las tuplas individualmente o todas a la vez mediante un checkbox especial llamado “Select all” cuya pulsación seleccionará todos los checkbox que tenga por debajo. Si una vez pulsado se vuelve a pulsar la casilla “Select all” todas las casillas que tenga por debajo se deseleccionarán. Esto se puede realizar individualmente también, pero cuando hay gran cantidad de información y se quiere borrar toda de una vez, la casilla “Select all” se hace casi necesaria. El checkbox select all se ha implementado como un evento que obliga a

poner el valor *true*, en el caso de la selección, o el valor *false*, en el caso de la deselección, a cada checkbox individual correspondiente. Existen, para cada conjunto de tuplas que se quieran borrar, tres eventos. Un evento corresponde a la selección/deselección de cada checkbox individual. Otro evento es el de la casilla de selección/deselección de todos los checkbox a la vez. Y el tercer evento es el de dar de baja a las casillas seleccionadas cuando se pulsa el botón “Accept”. Dentro de estos eventos se controla mediante booleanos y arrays de booleanos el estado anterior (*true* o *false*) para realizar el cambio correspondiente usando para ello el tipo de datos conocido como lista. Se almacena en una lista el conjunto de estados de los distintos checkbox y dependiendo de los eventos producidos se cambiarán estos estados.

El select all es un checkbox especial que obliga a realizar un refresco en la ventana para poder visualizar el cambio, ya que no es un checkbox especial de Java sino que se ha implementado para esta situación. Los checkbox normales de Java, cuando se pulsa sobre ellos, se puede visualizar el cambio en el momento, sin embargo en el caso del checkbox select all esto no ocurre y para poder visualizar el cambio de la selección o deselección se ha de realizar una actualización o refresco de la ventana, que consiste en crear de nuevo la ventana pero con una actualización o cambio en ella. Al realizar este refresco se puede observar el cambio pero en el caso de que se pulse un “Select all” de la parte más baja de la ventana, es decir que se pulse un checkbox select all que esté por debajo de la barra de desplazamiento vertical, el efecto será que se actualizará correctamente pero el scroll vertical volverá hacia arriba en la ventana obligando a tener que volver a bajar para poder pulsar el botón aceptar y visualizar el cambio recién hecho, lo cual es bastante incómodo. Por ello en esta clase se ha utilizado una variable de tipo *Point*, junto con métodos *View* de Java, que permite obtener el valor de las posiciones en la pantalla con el fin de que se pueda visualizar correctamente en cualquier punto en el que hagamos click en la casilla “Select all” y el scroll no se mueva al actualizar la pantalla impidiéndonos visualizar la información en el momento.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: “There are not X.”, donde X pueden ser problemas, palabras o sinónimos. Y si no hay información no aparecerá el botón de aceptar ni ninguna casilla checkbox de selección, solamente una etiqueta de texto *JLabel* informando del mensaje.

Cuando se pulsa sobre “Accept”, el botón se encuentra justo debajo de la última tupla de cada conjunto de información, la lista de estados que contengan el valor *true* se enviará a la base de datos mediante un bucle que realizará un borrado de todos esos valores mediante la instrucción *DELETE* y actualizará la ventana eliminando los datos recién borrados. Si se pulsa “Accept” sin seleccionar ninguna casilla la aplicación informará mediante un panel de aviso de que se debe seleccionar algún dato.

Todas las tuplas están representadas en una única línea, sin salto de línea, pero se pueden visualizar enteramente mediante la barra de desplazamiento horizontal. Esto se ha hecho así porque algunas tuplas contienen mucha información y si se hubiera representado en la ventana con saltos de línea habría sido menos legible a la hora de hacer la selección de las casillas.

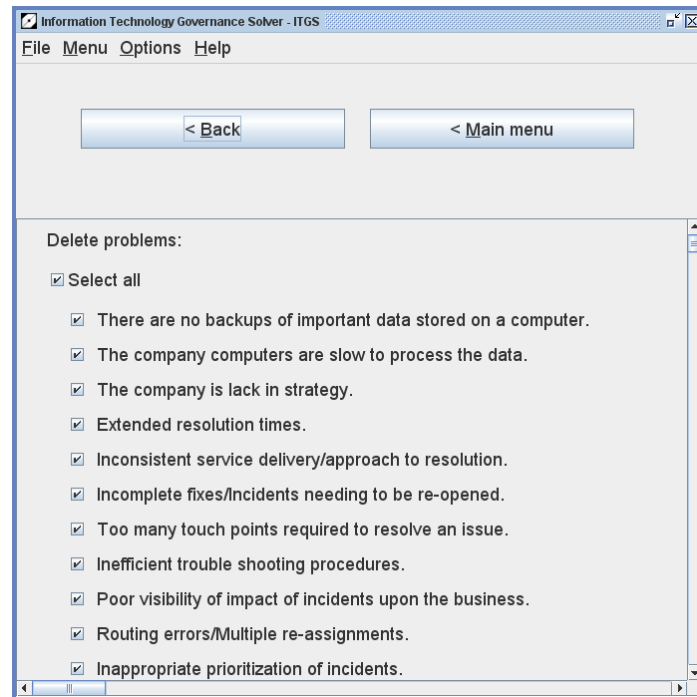


Ilustración 94. Baja en administración de problemas

La clase ***ventanaAdministrarProblemaModificacion.java*** permite realizar las modificaciones en la información de los problemas, palabras y sinónimos y la ventana es creada al pulsar sobre el botón “Update information”. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar Problema) y otro (“Main menu”) para ir al menú principal. Después se encuentran dos botones: “Change names” y “Relate database problems tables”. Al pulsar sobre el botón de cambiar nombres se lanza un evento que crea la clase ***ventanaAdministrarProblemaModificacionNombre.java*** y si se pulsa sobre el botón de relacionar las tablas de los problemas de la base de datos se lanza un evento que crea la clase ***ventanaAdministrarProblemaModificacionRelacion.java***.

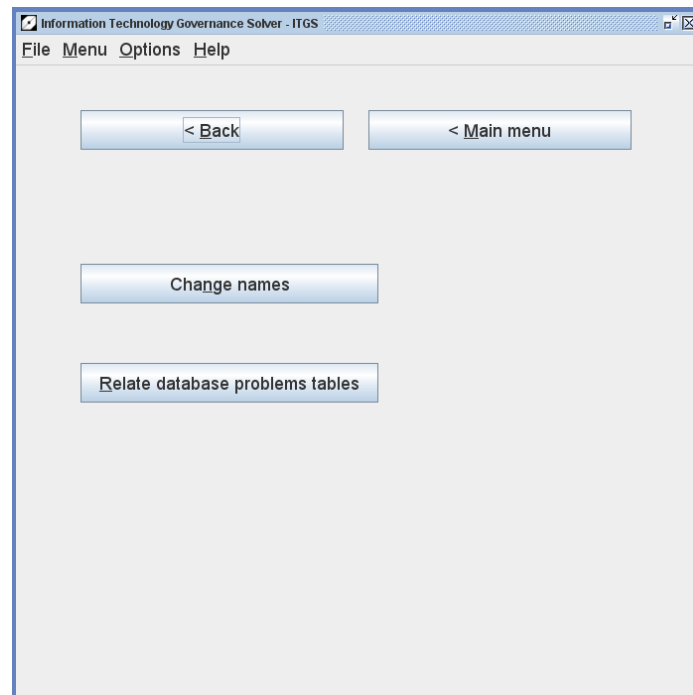


Ilustración 95. Modificación en administración de problemas

La llamada a la clase ***ventanaAdministrarProblemaModificacionNombre.java*** desde la ventana de modificación de problemas crea la ventana que permite cambiar los nombres o descripciones de la información en la base de datos de los problemas y sus palabras y sinónimos. En esta ventana existe un menú desplegable para cada conjunto de información, conjunto de tuplas de los problemas, las palabras y los sinónimos. Cuando se crea la ventana toda esta información se obtiene desde la base de datos y se representa por pantalla. La ventana posee una barra de desplazamiento vertical para poder visualizar la ventana completamente y consta de una caja de texto a la izquierda de cada menú desplegable y en la parte inferior izquierda de cada conjunto de información, justo debajo de cada caja de texto, de un botón de aceptar "Accept" para aplicar cambios. Los menús desplegables tienen sus propias barras de desplazamiento vertical y horizontal en caso de que la información sobrepase el tamaño del menú desplegable. La caja de texto de problema tiene un límite de 4000 caracteres, mientras que las cajas de texto de palabra y sinónimo tienen un límite de 2000 caracteres para la inserción de datos, ya que es así como está definido en la base de datos por las mismas razones que se comentaron en la clase de las altas.

Para cambiar el nombre o descripción de la información se debe seleccionar la tupla deseada del menú desplegable. Al hacer click sobre la tupla deseada, la caja de texto mostrará la información seleccionada para facilitar al administrador la tarea del cambio del nombre o descripción, para que en caso de que el administrador solo quisiera cambiar unos pocos caracteres en una tupla con mucha información, no tenga que reescribir todo de nuevo. En cualquier caso se puede borrar todo por si se quiere

escribir algo totalmente distinto. Para que funcione el efecto de que al pulsar sobre el menú desplegable, en la caja de texto aparezca el ítem del menú desplegable seleccionado, se necesita refrescar o actualizar la ventana para mostrar la información. Al realizar este refresco se puede observar el cambio pero en el caso de que se pulse un menú desplegable de la parte más baja de la ventana, es decir que se pulse un menú que esté por debajo de la barra de desplazamiento vertical, el efecto será que se actualizará correctamente pero el scroll vertical volverá hacia arriba en la ventana obligando a tener que volver a bajar para poder pulsar el botón aceptar y visualizar el cambio recién hecho, lo cual es bastante incómodo. Por ello en esta clase se ha utilizado una variable de tipo *Point*, más los métodos *View* de Java, que permite obtener el valor de las posiciones en la pantalla con el fin de que se pueda visualizar correctamente en cualquier punto en el que hagamos click y el scroll no se mueva al actualizar la pantalla impidiéndonos visualizar la información en el momento.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: "There are not X.", donde X pueden ser problemas, palabras o sinónimos. Y si no hay información no aparecerá el botón de aceptar ni el menú desplegable ni tampoco la caja de texto para introducir datos, solamente una etiqueta de texto *JLabel* informando del mensaje.

El funcionamiento de los eventos cuando se pulsa el botón aceptar es el siguiente:

Se realiza un tratamiento de la cadena introducida o modificada de la original de la caja de texto. Una vez libre de espacios no útiles (los espacios al principio o final de la cadena no aportan ninguna información) y de caracteres problemáticos de inyección SQL se procede a realizar una serie de acciones sobre la base de datos antes de insertarla. Pero antes de las acciones en la base de datos, lo primero es comprobar si la cadena nueva es igual que la original, en caso de ser iguales se avisa y no se realiza ningún cambio. En caso de que no sean iguales, se comprueba en la base de datos si ya existe el mismo nombre, en cuyo caso se avisa de que ya existe y no se inserta ni se cambia nada. En caso de que el nombre sea distinto se realizan acciones de backup, borrado e inserción. El cambio de nombre de una tupla supone el borrado de la original tupla e insertado de la nueva tupla en la base de datos. Para llevar a cabo esto primero se realizan unas copias de seguridad (backup) de todas las tablas intermedias que estén relacionadas con la tupla que se quiere modificar. Los backups se realizan mediante el tipo de datos lista, en la lista se recoge toda la información de los atributos de las tuplas. Una vez se tienen los backups se procede al borrado de la tupla original, como el borrado es en cascada no es necesario borrar la tupla original de las tablas intermedias, ya que al borrar en cascada borrará la tupla original de la tabla propia y de las tablas intermedias que estén involucradas. Una vez borrada la tupla original se inserta en su tabla la nueva tupla introducida en la caja de texto. Y a continuación con los backups que se hicieron en las listas obtenemos los atributos antiguos, generalmente el atributo código, para insertarlos con la nueva información en las tablas intermedias para que no se pierdan las relaciones entre tablas y la

información quede intacta pero con un nuevo nombre o descripción. Después de hacer las operaciones sobre la base de datos se hace una llamada para crear la ventana de nuevo y realizar el refresco o actualización con la nueva información de las cajas y de los menús.

Si se pulsa el botón “Accept” sin elegir información del menú desplegable el sistema nos lo avisará con un mensaje de aviso.

En la cuestión del tratamiento de los caracteres hay que tener en cuenta lo siguiente. La cadena que se introduce recibe un tratamiento de caracteres para evitar la inserción de caracteres especiales y evitar el problema de la inyección SQL. Algunas tuplas de la base de datos insertadas con los scripts tienen algunos de estos caracteres como son la comilla simple, la comilla doble y la barra invertida. Estos caracteres son visualizados correctamente tanto en el menú como en las cajas, pero si deseamos introducir estos caracteres en las áreas de introducción de texto, la aplicación no lo va a permitir.

Si introducimos un cambio en el texto sobre el texto original con caracteres no permitidos y pulsamos el botón aceptar, lo que va a suceder es que la aplicación va a avisar de que la tupla ya existe en la base de datos. Esto ocurre porque a efectos del programa los caracteres no permitidos no existen, ya que no se tienen en cuenta, y por tanto aunque se cambien o añadan caracteres no permitidos, al no tratarse, no se compararán, sino que se comprobará la cadena sin ellos y por tanto la cadena nueva y la antigua serán la misma y se encontrará en la base de datos la misma tupla por lo que no se insertará la nueva. Esto ocurre cuando la cadena original no tiene ningún carácter no permitido en su script.

Si la cadena original tiene uno o varios caracteres no permitidos, e introducimos un cambio en el texto sobre el original con caracteres no permitidos también, lo que va a ocurrir es que el sistema va a detectar que hay un cambio y lo va a introducir en la base de datos. Como hemos dicho antes, esto no debería pasar ya que los caracteres nuevos no permitidos que se introducen son desechados porque no se tienen en cuenta, pero lo que sucede es que la cadena original tenía caracteres no permitidos también, que a su vez van a ser ignorados, y como consecuencia, la cadena nueva va a ser una réplica de la original pero sin sus caracteres no permitidos, por lo que va a ser distinta de la original y se insertará en la base de datos.

Si introducimos caracteres permitidos tendrán un cambio normal en la inserción.

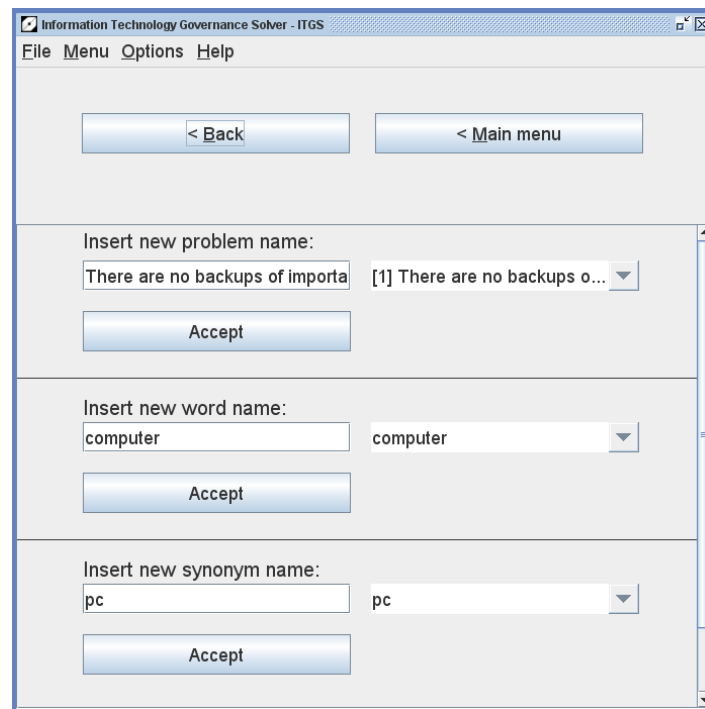


Ilustración 96. Modificación de nombres en administración de problemas

La llamada a la clase ***ventanaAdministrarProblemaModificacionRelacion.java*** desde la ventana de modificación de problemas crea la ventana que permite relacionar las tablas de la información en la base de datos de los problemas, palabras y sinónimos. Para que la solución tenga consistencia, los problemas deben estar relacionados con COBIT, y esta relación viene dada por proceso de COBIT. Un problema en la base de datos debe estar relacionado con uno o varios procesos de COBIT para que se encuentre posteriormente una relación en la solución. Además una o varias palabras deben estar relacionadas con las palabras que posee un problema y por último una o varias palabras pueden tener relación con uno o varios sinónimos. En esta ventana existen dos menús desplegables para cada tabla que se puede relacionar en la base de datos sobre los problemas (y procesos de COBIT), palabras y sinónimos. Las relaciones permitidas son las mismas relaciones existentes en la base de datos y son las siguientes:

- Problema con Proceso
- Palabra con Problema
- Palabra con Sinónimo

Cuando se crea la ventana toda esta información se obtiene desde la base de datos y se representa por pantalla en los menús desplegables.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: "There are not X.", donde X pueden ser problemas, procesos

de COBIT, palabras o sinónimos. Y si no hay información no aparecerá el botón de crear ni el de borrar relaciones, así como tampoco el menú desplegable correspondiente, solamente la etiqueta de texto JLabel informando del mensaje.

La ventana posee una barra de desplazamiento vertical para poder visualizar la ventana completamente y consta aparte de los menús desplegables, con sus propios scrolls vertical y horizontal en caso necesario, de dos botones para cada par de menús desplegables que permiten crear "Create relationship" o borrar "Delete relationship" una relación entre tablas.

Como se explicó anteriormente, la aplicación permite el insertado de información independiente y no restrictiva. Esto quiere decir que podemos introducir cualquier tipo de información pero se debe saber que en el alta la información no estará relacionada. Es por ello que en esta ventana de administración se permite relacionar las tablas de toda la información que exista en la base de datos acerca de los problemas, sus palabras y sus sinónimos, y su relación con COBIT a través del proceso.

Para crear o borrar una relación se debe seleccionar un ítem de los dos menús desplegables que se deseen relacionar o desrelacionar. En caso de pulsar uno de los botones disponibles y no seleccionar los dos ítems, el sistema lo indicará mediante un aviso.

En el caso de que se seleccionen ambos ítems de los menús y pulsemos el botón para crear la relación entre ellos, la aplicación comprueba en la base de datos si ambos ítems, que son tuplas, estaban ya relacionados o no. Esto se hace mediante una consulta a la tabla intermedia de ambas tuplas por atributo "código" (en palabra y sinónimo es su propia descripción) que es clave primaria, si la consulta contiene valor es que están relacionadas, si no se devuelve nada en la consulta es que no hay relación. Si ambas tuplas estaban relacionadas y pulsamos el botón de crear relación, el sistema nos indicará que esas tuplas ya tenían relación y no hará nada. En cambio si la aplicación comprueba en la tabla intermedia de ambas tuplas que no existía relación, al pulsar en el botón de crear, se creará la relación insertando los códigos de ambas tuplas en la tabla intermedia de ambas, creando de este modo la relación.

El proceso en el borrado de la relación es similar. Cuando se seleccionen ambas tuplas del par de menús desplegables se comprobará si existía o no una relación entre ellas mediante una consulta a la tabla intermedia. En caso de no existir relación entre las tuplas, al pulsar sobre el borrado de relación, la aplicación nos avisará de que no existe una relación previa y por tanto no hará nada. Pero si existía una relación anterior entre tuplas, la aplicación enviará a la base de datos una instrucción de borrado con los atributos correspondientes de ambas tablas de las dos tuplas para eliminar la tupla en la tabla intermedia, borrando de este modo la relación.

Como se puede observar, la tarea que se realiza en esta ventana, es la de inserción o borrado en la tabla intermedia que relaciona cada par de tablas de la base de datos.

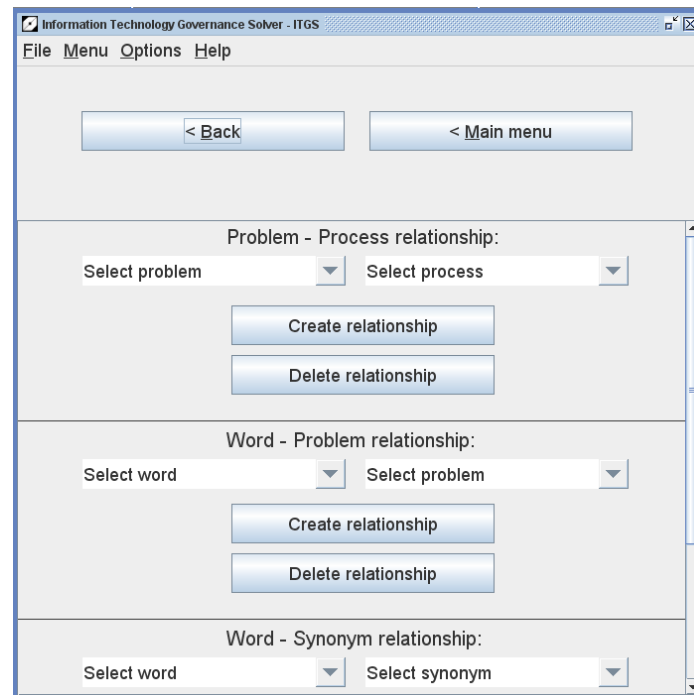


Ilustración 97. Relación de tablas en la administración de problemas

Configuración de la información de la evaluación:

Cuando pulsamos el botón “Configure assessments information” desde la “ventana Administrar” se crea la clase ***ventanaAdministrarEvaluacion.java*** que permite las tareas de dar de alta, de baja y de modificación mediante los botones “Add information”, “Delete information” y “Update information” respectivamente. También se permite volver hacia atrás mediante el botón “Back” con el que se regresaría a la “ventana Administrar” y se permite volver al menú principal mediante el botón “Main menu”.

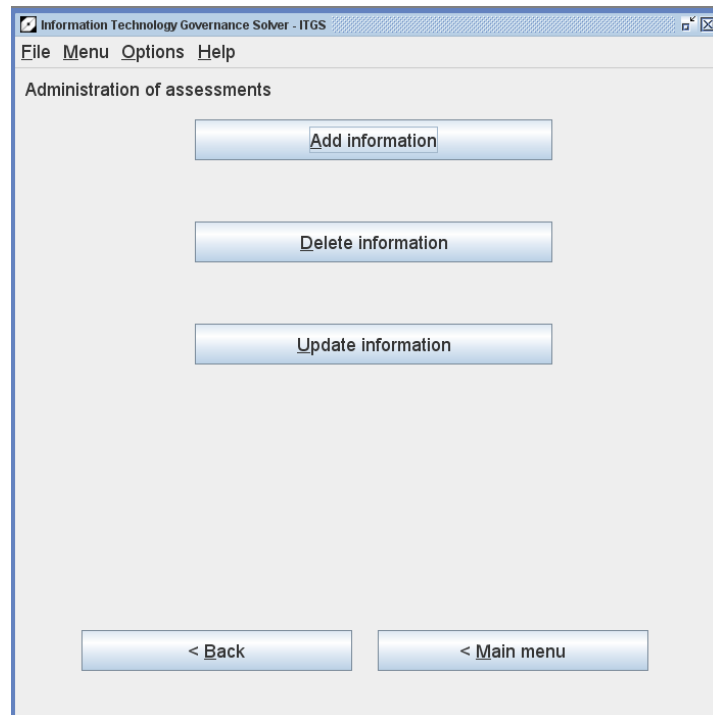


Ilustración 98. Administración de las evaluaciones

Cuando pulsemos cualquiera de los botones de las altas, bajas o modificaciones se crearán las siguientes clases para cada una de ellas. El nombre dado a las clases de la administración en la parte de información de la evaluación ha sido:

- Para las altas:

ventanaAdministrarEvaluacionAlta.java

- Para las bajas:

ventanaAdministrarEvaluacionBaja.java

- Para las modificaciones:

ventanaAdministrarEvaluacionModificacion.java

- Para el cambio de nombres:

ventanaAdministrarEvaluacionModificacionNombre.java

- Para la relación entre tablas:

ventanaAdministrarEvaluacionModificacionRelacion.java

La clase ***ventanaAdministrarEvaluacionAlta.java*** permite realizar las altas en la información de las evaluaciones y la ventana es creada al pulsar sobre el botón “Add information” desde la anterior clase ***ventanaAdministrarEvaluacion.java***. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar evaluaciones) y otro (“Main menu”) para ir al menú principal. Para dar de alta a una evaluación, se dispone de un área de texto con un límite de 4000 caracteres y con un scroll vertical en caso de que la información sea extensa para poder visualizarla completamente. La ventana dispone a su vez de una barra de desplazamiento vertical (scroll vertical) para poder visualizar toda la ventana entera sin problemas.

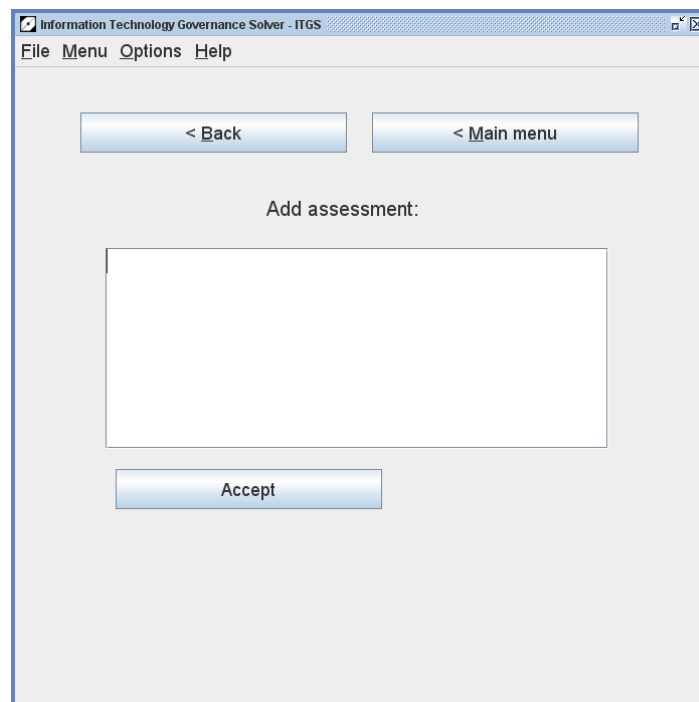


Ilustración 99. Alta en administración de evaluación

Cuando se pulsa en el botón justo debajo del área de texto (botón “Accept”) se realizan una serie de acciones que son las de tratamiento de la cadena introducida por

el administrador y una vez tratada se hace una inserción en la base de datos siempre que la información no sea repetida en la base de datos. Si se pulsa sin información introducida en la caja de texto el sistema nos indicará con un mensaje de que introduzcamos información válida. Cuando existe información, el tratamiento de la cadena se ocupa de quitar los espacios que no aportan información del comienzo y del final, y elimina todos los signos que pudieran provocar una inyección SQL, permitiendo solo el uso de algunos caracteres especiales, por lo que si solo se introducen caracteres no permitidos el sistema nos avisará de que se introduzcan valores válidos. También se comprueba que no sea una cadena vacía y todo ello mediante mensajes de aviso. Cuando la cadena de caracteres está libre de posibles caracteres problemáticos se realiza un aumento en el código de la tupla de la base de datos, teniendo en cuenta el valor máximo de los códigos ya existentes y añadiéndole un número más de incremento. En el caso de que no hubiera tuplas, no existiría valor máximo y simplemente en caso de ser la primera tupla insertada se iniciaría el código con el valor 1. Antes justo de la inserción, se comprueba que este número incrementado no supere el valor máximo existente en la base de datos, en cuyo caso se avisaría, y en caso de no superarlo se comprueba que la información nueva no exista ya en la base de datos. Si existe ya, no se inserta y se avisa con un mensaje, y si no existe se hace la inserción en la base de datos.

La clase ***ventanaAdministrarEvaluacionBaja.java*** permite realizar las bajas en la información de las evaluaciones y la ventana es creada al pulsar sobre el botón “Delete information”. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar Evaluaciones) y otro (“Main menu”) para ir al menú principal. Después se encuentran una serie de cuadrados seleccionables o checkbox que permiten la selección de varios elementos a la vez. La ventana posee una barra de desplazamiento vertical y horizontal para visualizar todo el contenido correctamente. En esta ventana aparece toda la información existente sobre las evaluaciones. Cuando se crea la ventana toda la información se obtiene desde la base de datos y se representa por pantalla.

Se pueden seleccionar o deseleccionar cada una de las tuplas individualmente o todas a la vez mediante un checkbox especial llamado “Select all” cuya pulsación seleccionará todos los checkbox que tenga por debajo. Si una vez pulsado se vuelve a pulsar la casilla “Select all” todas las casillas que tenga por debajo se deseleccionarán. Esto se puede realizar individualmente también, pero cuando hay gran cantidad de información y se quiere borrar toda de una vez, la casilla “Select all” se hace casi necesaria. El checkbox select all se ha implementado como un evento que obliga a poner el valor *true*, en el caso de la selección, o el valor *false*, en el caso de la desección, a cada checkbox individual correspondiente. Existen tres eventos que manejan el tratamiento de los checkbox. Un evento corresponde a la selección/desección de cada checkbox individual. Otro evento es el de la casilla de selección/desección de todos los checkbox a la vez. Y el tercer evento es el de dar de baja a las casillas seleccionadas cuando se pulsa el botón “Accept”. Dentro de estos eventos, se controla mediante booleanos y arrays de booleanos el estado anterior

(*true* o *false*) para realizar el cambio correspondiente usando para ello el tipo de datos conocido como lista. Se almacena en una lista el conjunto de estados de los distintos checkbox y dependiendo de los eventos producidos se cambiarán estos estados.

El select all es un checkbox especial que obliga a realizar un refresco en la ventana para poder visualizar el cambio, ya que no es un checkbox especial de Java sino que se ha implementado para esta situación. Los checkbox normales de Java, cuando se pulsa sobre ellos, se puede visualizar el cambio en el momento, sin embargo en el caso del checkbox select all esto no ocurre y para poder visualizar el cambio de la selección o desección se ha de realizar una actualización o refresco de la ventana, que consiste en crear de nuevo la ventana pero con una actualización o cambio en ella, por ejemplo el cambio podría ser poner todos los checkbox a true, es decir seleccionados, o a false, deseleccionados, dependiendo del uso del Select all. Cuando se pulsa un select all que estuviera por ejemplo en una parte más baja de la ventana, al realizar este refresco se actualizará correctamente pero el scroll vertical volverá hacia arriba en la ventana obligando a tener que volver a bajar para poder pulsar el botón aceptar y visualizar el cambio recién hecho, lo cual es incómodo. Pero como esta clase tan solo tiene un Select all ya que la única información que se puede borrar es referente a la misma tabla de evaluaciones, no habría ningún problema porque no se puede hacer un select all en la parte más baja puesto que no hay más que uno. Sin embargo, si se mueve un poco el scroll, pero pudiendo ver la casilla select all, notaríamos en el refresco que la ventana subiría un poco su scroll, justo el contrario del que movimos hacia abajo. Por ello y aunque no es muy necesario y por seguir el mismo sistema que en las demás ventanas, en esta clase se ha utilizado también una variable de tipo *Point*, junto con métodos *View* de Java, que permite obtener el valor de las posiciones en la pantalla con el fin de que se pueda visualizar correctamente en cualquier punto en el que hagamos click en la casilla "Select all" y el scroll no se mueva al actualizar la pantalla impidiéndonos visualizar la información en el momento.

Si no existe información sobre ninguna evaluación, se verá reflejado en la pantalla con el mensaje: "There are not assessments.". Y si no hay información no aparecerá el botón de aceptar ni ninguna casilla checkbox de selección, solamente una etiqueta de texto *JLabel* informando del mensaje.

Cuando se pulsa sobre "Accept", el botón aparece justo debajo de la última tupla de la base de datos, la lista de estados que contengan el valor true se enviará a la base de datos mediante un bucle que realizará un borrado de todos esos valores mediante la instrucción *DELETE* y actualizará la ventana eliminando los datos recién borrados. Si se pulsa "Accept" sin seleccionar ninguna casilla la aplicación informará mediante un panel de aviso de que se debe seleccionar algún dato.

Todas las tuplas están representadas en una única línea, sin salto de línea, pero se pueden visualizar enteramente mediante la barra de desplazamiento horizontal. Esto se ha hecho así porque algunas tuplas contienen mucha información y si se

hubiera representado en la ventana con saltos de línea habría sido menos legible a la hora de hacer la selección de las casillas.

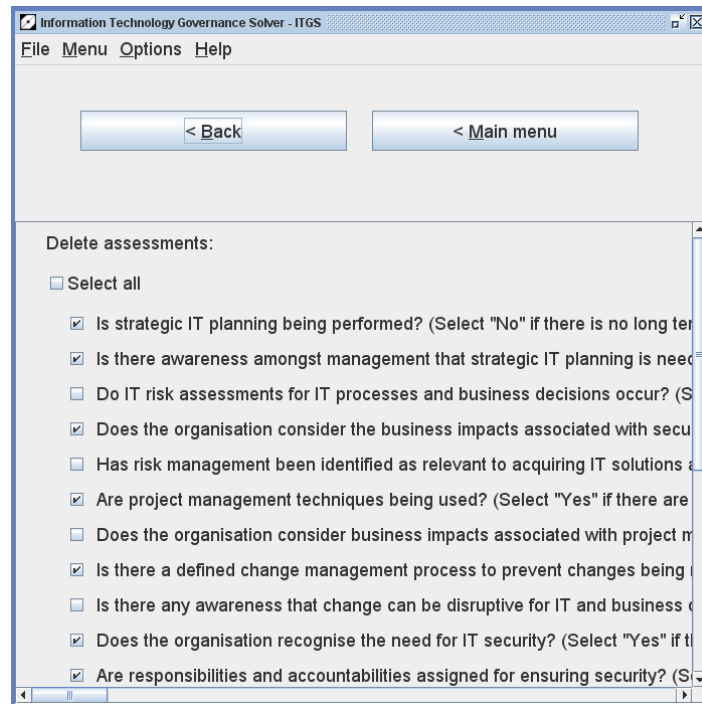


Ilustración 100. Baja en administración de evaluaciones

La clase **ventanaAdministrarEvaluacionModificacion.java** permite realizar las modificaciones en la información de las evaluaciones y la ventana es creada al pulsar sobre el botón “Update information”. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar Evaluaciones) y otro (“Main menu”) para ir al menú principal. Después se encuentran dos botones: “Change names” y “Relate database assessments tables”. Al pulsar sobre el botón de cambiar nombres se lanza un evento que crea la clase **ventanaAdministrarEvaluacionModificacionNombre.java** y si se pulsa sobre el botón de relacionar las tablas de las evaluaciones de la base de datos se lanza un evento que crea la clase **ventanaAdministrarEvaluacionModificacionRelacion.java**.

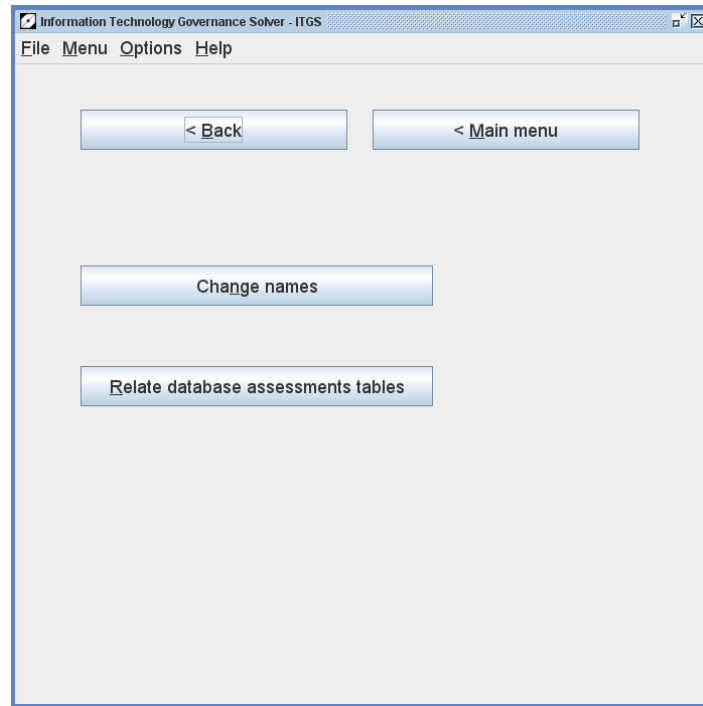


Ilustración 101. Modificación en administración de Evaluación

La llamada a la clase ***ventanaAdministrarEvaluacionModificacionNombre.java*** desde la ventana de modificación de evaluaciones crea la ventana que permite cambiar los nombres o descripciones de la información en la base de datos de las evaluaciones. En esta ventana existe un solo menú desplegable para el conjunto de tuplas pertenecientes a las evaluaciones. Cuando se crea la ventana toda la información se obtiene desde la base de datos y se representa por pantalla. La ventana consta de una caja de texto a la izquierda del menú desplegable y en la parte inferior izquierda, justo debajo de la caja de texto, consta de un botón de aceptar "Accept" para aplicar cambios. El menú desplegable tiene sus propias barras de desplazamiento vertical y horizontal en caso de que la información sobrepase el tamaño del menú desplegable. La caja de texto de la evaluación tiene un límite de 4000 caracteres.

Para cambiar el nombre o descripción de la evaluación se debe seleccionar la tupla deseada del menú desplegable. Al hacer click sobre la tupla deseada, la caja de texto mostrará la información seleccionada para facilitar al administrador la tarea del cambio del nombre o descripción, para que en caso de que el administrador solo quisiera cambiar unos pocos caracteres en una tupla con mucha información, no tenga que reescribir todo de nuevo. En cualquier caso se puede borrar todo por si se quiere escribir algo totalmente distinto. Para que funcione el efecto de que al pulsar sobre el menú desplegable, en la caja de texto aparezca el ítem del menú desplegable seleccionado, se necesita refrescar o actualizar la ventana para mostrar la información sobre la caja de texto.

Si no existe información sobre las evaluaciones se verá reflejado en la pantalla con el mensaje: "There are not assessments.". Y si no hay información no aparecerá el botón de aceptar ni el menú desplegable ni tampoco la caja de texto para introducir datos, solamente una etiqueta de texto JLabel informando del mensaje.

El funcionamiento de los eventos cuando se pulsa el botón aceptar es el siguiente:

Se realiza un tratamiento de la cadena introducida o modificada de la original de la caja de texto. Una vez libre de espacios no útiles (los espacios al principio o final de la cadena no aportan ninguna información) y de caracteres problemáticos de inyección SQL se procede a realizar una serie de acciones sobre la base de datos antes de insertarla. Pero antes de las acciones en la base de datos, lo primero es comprobar si la cadena nueva es igual que la original, en caso de ser iguales se avisa y no se realiza ningún cambio. En caso de que no sean iguales, se comprueba en la base de datos si ya existe el mismo nombre, en cuyo caso se avisa de que ya existe y no se inserta ni se cambia nada. En caso de que el nombre sea distinto se realizan acciones de backup, borrado e inserción. El cambio de nombre de una tupla supone el borrado de la original tupla e insertado de la nueva tupla en la base de datos. Para llevar a cabo esto primero se realizan unas copias de seguridad (backup) de todas las tablas intermedias que estén relacionadas con la tupla que se quiere modificar. Los backups se realizan mediante el tipo de datos lista, en la lista se recoge toda la información de los atributos de las tuplas. Una vez se tienen los backups se procede al borrado de la tupla original, como el borrado es en cascada no es necesario borrar la tupla original de las tablas intermedias, ya que al borrar en cascada borrará la tupla original de la tabla propia y de las tablas intermedias que estén involucradas. Una vez borrada la tupla original se inserta en su tabla la nueva tupla introducida en la caja de texto. Y a continuación con los backups que se hicieron en las listas obtenemos los atributos antiguos, generalmente el atributo código, para insertarlos con la nueva información en las tablas intermedias para que no se pierdan las relaciones entre tablas y la información quede intacta pero con un nuevo nombre o descripción. Después de hacer las operaciones sobre la base de datos se hace una llamada para crear la ventana de nuevo y realizar el refresco o actualización con la nueva información de la caja y del menú.

Si se pulsa el botón "Accept" sin elegir información del menú desplegable el sistema nos lo avisará con un mensaje de aviso.

En la cuestión del tratamiento de los caracteres hay que tener en cuenta lo siguiente. La cadena que se introduce recibe un tratamiento de caracteres para evitar la inserción de caracteres especiales y evitar el problema de la inyección SQL. Algunas tuplas de la base de datos insertadas con los scripts tienen algunos de estos caracteres como son la comilla simple, la comilla doble y la barra invertida. Estos caracteres son visualizados correctamente tanto en el menú como en las cajas, pero si deseamos

introducir estos caracteres en las áreas de introducción de texto, la aplicación no lo va a permitir.

Si introducimos un cambio en el texto sobre el texto original con caracteres no permitidos y pulsamos el botón aceptar, lo que va a suceder es que la aplicación va a avisar de que la tupla ya existe en la base de datos. Esto ocurre porque a efectos del programa los caracteres no permitidos no existen, ya que no se tienen en cuenta, y por tanto aunque se cambien o añadan caracteres no permitidos, al no tratarse, no se compararán, sino que se comprobará la cadena sin ellos y por tanto la cadena nueva y la antigua serán la misma y se encontrará en la base de datos la misma tupla por lo que no se insertará la nueva. Esto ocurre cuando la cadena original no tiene ningún carácter no permitido en su script.

Si la cadena original tiene uno o varios caracteres no permitidos, e introducimos un cambio en el texto sobre el original con caracteres no permitidos también, lo que va a ocurrir es que el sistema va a detectar que hay un cambio y lo va a introducir en la base de datos. Como hemos dicho antes, esto no debería pasar ya que los caracteres nuevos no permitidos que se introducen son desechados porque no se tienen en cuenta, pero lo que sucede es que la cadena original tenía caracteres no permitidos también, que a su vez van a ser ignorados, y como consecuencia, la cadena nueva va a ser una réplica de la original pero sin sus caracteres no permitidos, por lo que va a ser distinta de la original y se insertará en la base de datos.

Si introducimos caracteres permitidos tendrán un cambio normal en la inserción.

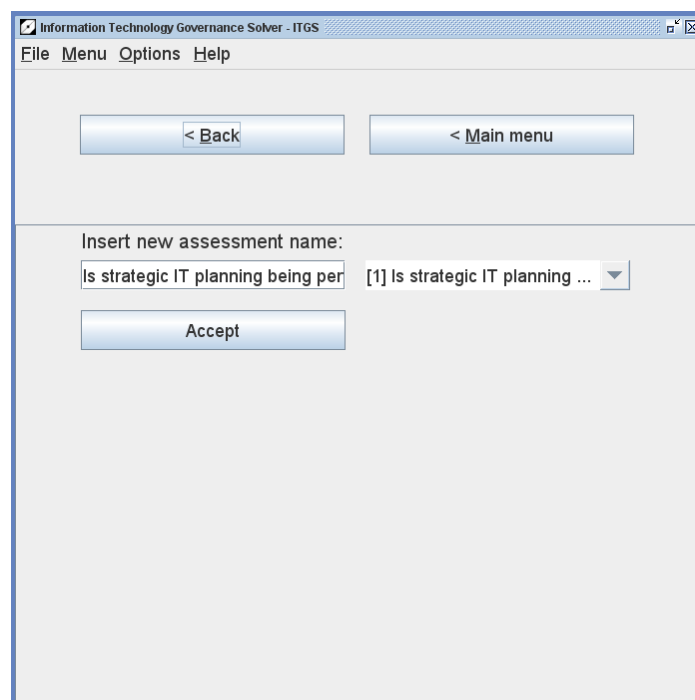


Ilustración 102. Modificación de nombre en administración de evaluaciones

La llamada a la clase ***ventanaAdministrarEvaluacionModificacionRelacion.java*** desde la ventana de modificación de evaluaciones crea la ventana que permite relacionar las tablas de la información en la base de datos de las evaluaciones. Para que la solución tenga consistencia, las evaluaciones deben estar relacionadas con COBIT, y esta relación viene dada por proceso de COBIT. Una evaluación en la base de datos debe estar relacionada con uno o varios procesos de COBIT para que se encuentre posteriormente una madurez por la rama del usuario de las evaluaciones y una relación en la solución. En esta ventana existen dos menús desplegables para poder relacionar una evaluación con un proceso de COBIT. Por tanto la única relación permitida entre tablas es la siguiente:

- Evaluación con Proceso

Cuando se crea la ventana, toda la información acerca de evaluaciones y procesos se obtiene desde la base de datos y se representa por pantalla en los menús desplegables.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: “There are not X.”, donde X pueden ser evaluaciones o procesos de COBIT. Y si no hay información no aparecerá el botón de crear ni el de borrar relaciones, así como tampoco el menú desplegable correspondiente, solamente la etiqueta de texto JLabel informando del mensaje.

La ventana consta aparte de los menús desplegables, con sus propios scrolls vertical y horizontal en caso necesario, de dos botones para el par de menús desplegables que permiten crear “Create relationship” o borrar “Delete relationship” una relación entre tablas.

Como se explicó anteriormente, la aplicación permite el insertado de información independiente y no restrictiva. Esto quiere decir que podemos introducir cualquier tipo de información pero se debe saber que en el alta la información no estará relacionada. Es por ello que en esta ventana de administración se permite relacionar las tablas de toda la información que exista en la base de datos acerca de las evaluaciones y su relación con COBIT a través del proceso.

Para crear o borrar una relación se debe seleccionar un ítem de los dos menús desplegables que se deseen relacionar o desrelacionar. En caso de pulsar uno de los botones disponibles y no seleccionar los dos ítems, el sistema lo indicará mediante un aviso.

En el caso de que se seleccionen ambos ítems de los menús y pulsemos el botón para crear la relación entre ellos, la aplicación comprueba en la base de datos si ambos ítems, que son tuplas, estaban ya relacionados o no. Esto se hace mediante una consulta a la tabla intermedia de ambas tuplas por atributo “código” que es clave primaria, si la consulta contiene valor es que están relacionadas, si no se devuelve

nada en la consulta es que no hay relación. Si ambas tuplas estaban relacionadas y pulsamos el botón de crear relación, el sistema nos indicará que esas tuplas ya tenían relación y no hará nada. En cambio si la aplicación comprueba en la tabla intermedia de ambas tuplas que no existía relación, al pulsar en el botón de crear, se creará la relación insertando los códigos de ambas tuplas en la tabla intermedia de ambas, creando de este modo la relación.

El proceso en el borrado de la relación es similar. Cuando se seleccionen ambas tuplas del par de menús desplegables se comprobará si existía o no una relación entre ellas mediante una consulta a la tabla intermedia. En caso de no existir relación entre las tuplas, al pulsar sobre el borrado de relación, la aplicación nos avisará de que no existe una relación previa y por tanto no hará nada. Pero si existía una relación anterior entre tuplas, la aplicación enviará a la base de datos una instrucción de borrado con los atributos código de ambas tablas de las dos tuplas para eliminar la tupla en la tabla intermedia, borrando de este modo la relación.

Como se puede observar, la tarea que se realiza en esta ventana, es la de inserción o borrado en la tabla intermedia que relaciona el par de tablas de la base de datos.

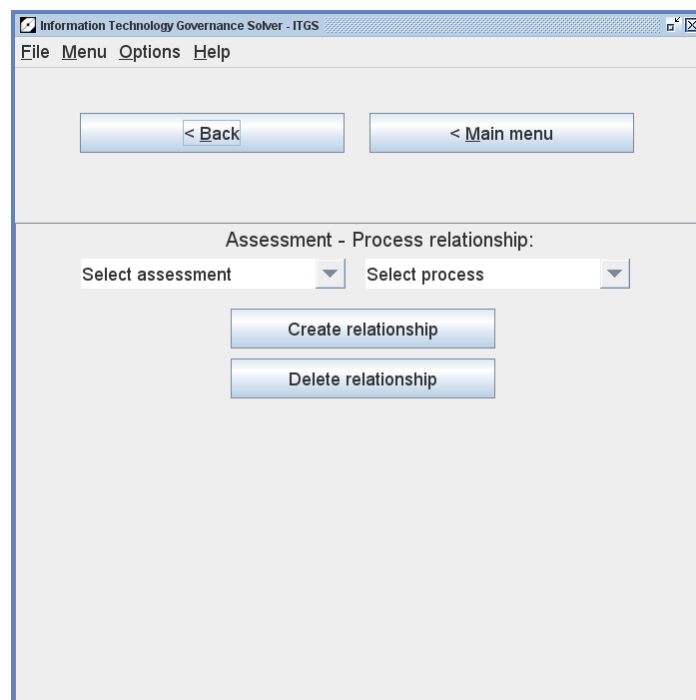


Ilustración 103. Relación de tablas en la administración de evaluaciones

Configuración de la información de COBIT:

Cuando pulsamos el botón “Configure COBIT information” desde la “ventana Administrar” se crea la clase ***ventanaAdministrarCobit.java*** que permite las tareas de dar de alta, de baja y de modificación mediante los botones “Add information”, “Delete information” y “Update information” respectivamente. También se permite volver hacia atrás mediante el botón “Back” con el que se regresaría a la “ventana Administrar” y se permite volver al menú principal mediante el botón “Main menu”.

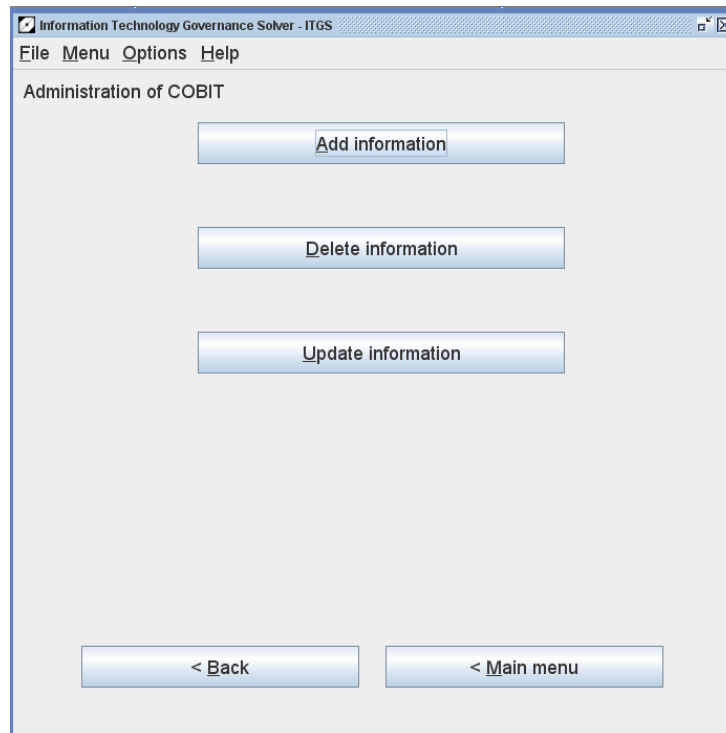


Ilustración 104. Administración de COBIT

Cuando pulsemos cualquiera de los botones de las altas, bajas o modificaciones se crearán las siguientes clases para cada una de ellas. El nombre dado a las clases de la administración en la parte de información de COBIT ha sido:

- Para las altas:

ventanaAdministrarCobitAlta.java

- Para las bajas:

ventanaAdministrarCobitBaja.java

- Para las modificaciones:

ventanaAdministrarCobitModificacion.java

- Para el cambio de nombres:

ventanaAdministrarCobitModificacionNombre.java

- Para la relación entre tablas:

ventanaAdministrarCobitModificacionRelacion.java

La clase ***ventanaAdministrarCobitAlta.java*** permite realizar las altas en la información de COBIT y la ventana es creada al pulsar sobre el botón “Add information”. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar COBIT) y otro (“Main menu”) para ir al menú principal. Después se encuentran una serie de cajas de texto para añadir toda la información que puede existir en la base de datos sobre COBIT. La ventana dispone de una barra de desplazamiento vertical (scroll vertical) para poder visualizar toda la ventana entera sin problemas. Cada caja de texto tiene un límite de 4000 caracteres excepto la caja “valor” cuyo límite es de 1 carácter numérico ya que solo se permite un valor entre 0 y 5. La información que se puede introducir de COBIT es toda la que existe en la base de datos que es la siguiente:

1. Dominio
2. Proceso
3. Objetivo de control
4. Objetivo TI
5. Objetivo de negocio
6. Meta
7. Métrica
8. Acción
9. Modelo de madurez y su valor

Ilustración 105. Alta en administración de COBIT

Cuando se pulsa en el botón correspondiente justo debajo de cada caja de texto (botón “Accept”) se realizan una serie de acciones que son las de tratamiento de la cadena introducida por el administrador y una vez tratada se hace una inserción en la base de datos siempre que la información no sea repetida en la base de datos. Si se pulsa sin información introducida en la caja de texto el sistema nos indicará con un mensaje de que introduzcamos información válida. Cuando existe información, el tratamiento de la cadena se ocupa de quitar los espacios que no aportan información del comienzo y del final, y elimina todos los signos que pudieran provocar una inyección SQL, permitiendo solo el uso de algunos caracteres especiales, por lo que si solo se introducen caracteres no permitidos el sistema nos avisará de que se introduzcan valores válidos. También se comprueba que no sea una cadena vacía y todo ello mediante mensajes de aviso. Cuando la cadena de caracteres está libre de posibles caracteres problemáticos se realiza un aumento en el código de la tupla de la base de datos, teniendo en cuenta el valor máximo de los códigos ya existentes y añadiéndole un número más de incremento. En el caso de que no hubiera tuplas, no existiría valor máximo y simplemente en caso de ser la primera tupla insertada se iniciaría el código con el valor 1. Antes justo de la inserción, se comprueba que este número incrementado no supere el valor máximo existente en la base de datos, en cuyo caso se avisaría, y en caso de no superarlo se comprueba que la información nueva no exista ya en la base de datos. Si existe ya, no se inserta y se avisa con un mensaje, y si no existe se hace la inserción en la base de datos. Todas las cajas de texto realizan las mismas acciones con la excepción de la caja del modelo de madurez que siempre debe llevar asociado un valor numérico que debe estar entre 0 y 5 ambos incluidos. Así mismo si introducimos el valor primero, este debe llevar asociado una descripción de modelo de madurez. Por lo tanto en esta parte del modelo de madurez se controla que estos datos se introduzcan y sean los correctos.

La clase ***ventanaAdministrarCobitBaja.java*** permite realizar las bajas en la información de COBIT y la ventana es creada al pulsar sobre el botón “Delete information”. La ventana contiene dos botones en la parte superior, uno (“Back”) para volver hacia la ventana inmediatamente anterior (ventana administrar COBIT) y otro (“Main menu”) para ir al menú principal. Después se encuentran una serie de cuadrados seleccionables o checkbox que permiten la selección de varios elementos a la vez. La ventana posee una barra de desplazamiento vertical y horizontal para visualizar todo el contenido correctamente. En esta ventana aparece toda la información existente de COBIT, esto son los dominios, procesos, objetivos de control, objetivos TI, objetivos de negocio, metas, métricas, acciones y modelos de madurez. Cuando se crea la ventana toda esta información se obtiene desde la base de datos y se representa por pantalla.

Se pueden seleccionar o deseleccionar cada una de las tuplas individualmente o todas a la vez mediante un checkbox especial llamado “Select all” cuya pulsación seleccionará todos los checkbox que tenga por debajo. Si una vez pulsado se vuelve a pulsar la casilla “Select all” todas las casillas que tenga por debajo se deseleccionarán. Esto se puede realizar individualmente también, pero cuando hay gran cantidad de

información y se quiere borrar toda de una vez, la casilla “Select all” se hace casi necesaria. El checkbox select all se ha implementado como un evento que obliga a poner el valor *true*, en el caso de la selección, o el valor *false*, en el caso de la deselección, a cada checkbox individual correspondiente. Existen, para cada conjunto de tuplas que se quieran borrar, tres eventos. Un evento corresponde a la selección/deselección de cada checkbox individual. Otro evento es el de la casilla de selección/deselección de todos los checkbox a la vez. Y el tercer evento es el de dar de baja a las casillas seleccionadas cuando se pulsa el botón “Accept”. Dentro de estos eventos se controla mediante booleanos y arrays de booleanos el estado anterior (*true* o *false*) para realizar el cambio correspondiente usando para ello el tipo de datos conocido como lista. Se almacena en una lista el conjunto de estados de los distintos checkbox y dependiendo de los eventos producidos se cambiarán estos estados.

El select all es un checkbox especial que obliga a realizar un refresco en la ventana para poder visualizar el cambio, ya que no es un checkbox especial de Java sino que se ha implementado para esta situación. Los checkbox normales de Java, cuando se pulsa sobre ellos, se puede visualizar el cambio en el momento, sin embargo en el caso del checkbox select all esto no ocurre y para poder visualizar el cambio de la selección o deselección se ha de realizar una actualización o refresco de la ventana, que consiste en crear de nuevo la ventana pero con una actualización o cambio en ella. Al realizar este refresco se puede observar el cambio pero en el caso de que se pulse un “Select all” de la parte más baja de la ventana, es decir que se pulse un checkbox select all que esté por debajo de la barra de desplazamiento vertical, el efecto será que se actualizará correctamente pero el scroll vertical volverá hacia arriba en la ventana obligando a tener que volver a bajar para poder pulsar el botón aceptar y visualizar el cambio recién hecho, lo cual es bastante incómodo. Por ello en esta clase se ha utilizado una variable de tipo *Point*, junto con métodos *View* de Java, que permite obtener el valor de las posiciones en la pantalla con el fin de que se pueda visualizar correctamente en cualquier punto en el que hagamos click en la casilla “Select all” y el scroll no se mueva al actualizar la pantalla impidiéndonos visualizar la información en el momento.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: “There are not X.”, donde X pueden ser dominios, procesos, objetivos de control... cualquier información que no esté en la base de datos. Y si no hay información no aparecerá el botón de aceptar ni ninguna casilla checkbox de selección, solamente una etiqueta de texto *JLabel* informando del mensaje.

Cuando se pulsa sobre “Accept”, el botón se encuentra justo debajo de la última tupla de cada conjunto de información, la lista de estados que contengan el valor *true* se enviará a la base de datos mediante un bucle que realizará un borrado de todos esos valores mediante la instrucción *DELETE* y actualizará la ventana eliminando los datos recién borrados. Si se pulsa “Accept” sin seleccionar ninguna casilla la aplicación informará mediante un panel de aviso de que se debe seleccionar algún dato.

Todas las tuplas están representadas en una única línea, sin salto de línea, pero se pueden visualizar enteramente mediante la barra de desplazamiento horizontal. Esto se ha hecho así porque algunas tuplas contienen mucha información y si se hubiera representado en la ventana con saltos de línea habría sido menos legible a la hora de hacer la selección de las casillas.

El conjunto de tuplas perteneciente al conjunto de información de los objetivos de control es un caso especial. Se puede observar que la información en este caso aparece con algunos caracteres juntos cuando deberían estar separados. Este es un problema que tiene la base de datos SQL de Oracle con Java en la representación que hacen de los saltos de líneas. Los objetivos de control son las tuplas que más información contienen en la base de datos por lo que se hizo necesario realizar saltos de líneas en la información y en algunos casos una inserción especial mediante procedimiento especial de inserción, no mediante la típica instrucción de insertado *INSERT*, ya que superan los límites de caracteres que SQL permite. Estos saltos de línea son los que en la ventana hacen que los caracteres aparezcan juntos de una línea y la siguiente porque Java no permite realizar un salto de línea en un JLabel (etiqueta de texto). Esto podría solucionarse mediante el añadido de un espacio más el carácter de retorno de carro (en código ASCII corresponde al valor 13, CHR (13)) en la base de datos después de cada salto de línea pero no tiene sentido ya que se estaría modificando la información cuando es correcta y además en la solución que se muestra por pantalla y en el guardado en el fichero esta información aparece correctamente ya que tienen un tratamiento distinto. Otra solución habría sido la de utilizar diversos JLabels para cada línea y mostrarla en distintas líneas pero esto haría como hemos comentado anteriormente una ventana menos legible para el seleccionado de los distintos checkbox. Otra solución habría sido realizar un tratamiento como el que se hace en la solución o en el guardado en el fichero, que es un tratado de palabra por palabra, pero habría complicado en exceso el código, sobre todo cuando se trata de un problema de la interpretación de los saltos de línea de Java. En la forma actual se hace más cómoda la lectura de la información. Si se introducen nuevos objetivos de control desde el alta del programa, este problema ya no ocurrirá puesto que el límite es de 4000 caracteres y se introducirán con una inserción normal sin saltos de línea.

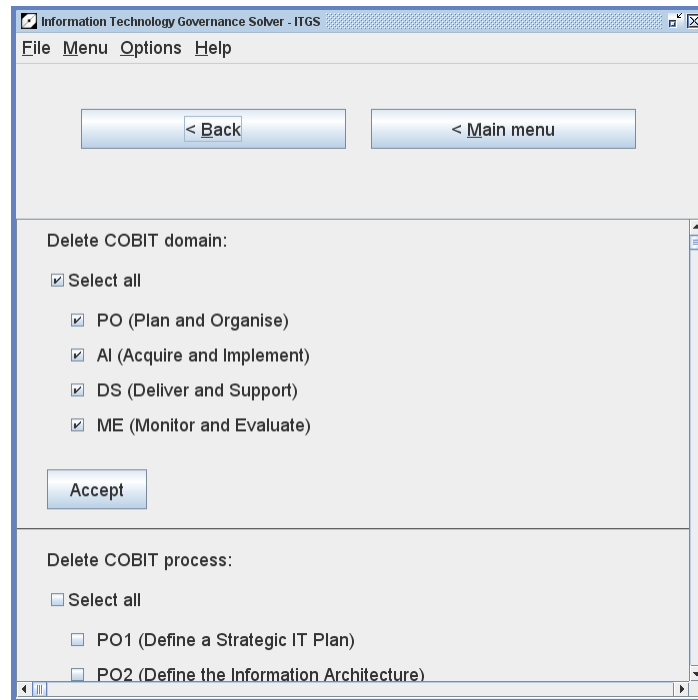


Ilustración 106. Baja en administración de COBIT

La clase ***ventanaAdministrarCobitModificacion.java*** permite realizar las modificaciones en la información de COBIT y la ventana es creada al pulsar sobre el botón "Update information". La ventana contiene dos botones en la parte superior, uno ("Back") para volver hacia la ventana inmediatamente anterior (ventana administrar COBIT) y otro ("Main menu") para ir al menú principal. Después se encuentran dos botones: "Change names" y "Relate database Cobit tables". Al pulsar sobre el botón de cambiar nombres se lanza un evento que crea la clase ***ventanaAdministrarCobitModificacionNombre.java*** y si se pulsa sobre el botón de relacionar las tablas de Cobit de la base de datos se lanza un evento que crea la clase ***ventanaAdministrarCobitModificacionRelacion.java***.

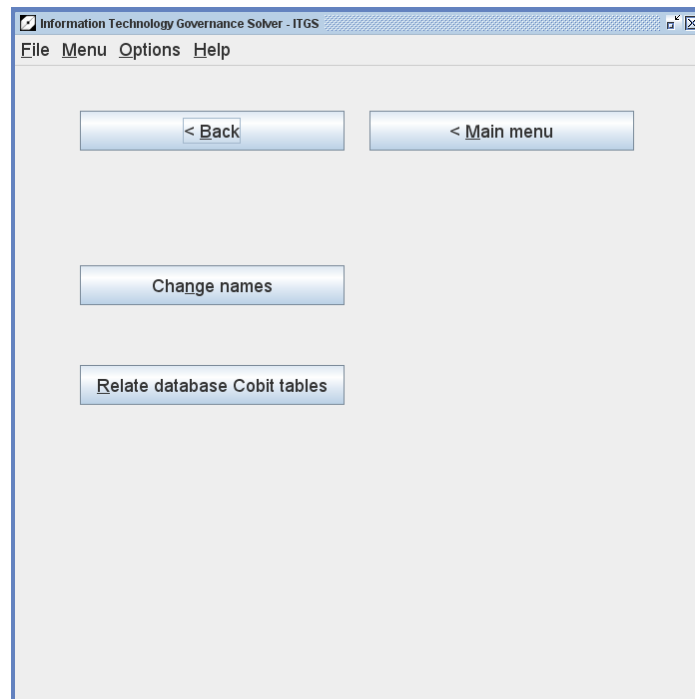


Ilustración 107. Modificación en administración de COBIT

La llamada a la clase ***ventanaAdministrarCobitModificacionNombre.java*** desde la ventana de modificación de Cobit crea la ventana que permite cambiar los nombres o descripciones de la información en la base de datos de COBIT. En esta ventana existe un menú desplegable por cada dominio, proceso, objetivo de control, objetivo TI, objetivo de negocio, meta, métrica, acción, modelo de madurez y su valor. Es decir un menú desplegable por cada conjunto de tuplas de toda la información disponible de COBIT. Cuando se crea la ventana toda esta información se obtiene desde la base de datos y se representa por pantalla. La ventana posee una barra de desplazamiento vertical para poder visualizar la ventana completamente y consta de una caja de texto a la izquierda de cada menú desplegable y en la parte inferior izquierda de cada conjunto de información, justo debajo de cada caja de texto, de un botón de aceptar "Accept" para aplicar cambios. Los menús desplegables tienen sus propias barras de desplazamiento vertical y horizontal en caso de que la información sobrepase el tamaño del menú desplegable. Las cajas de texto tienen un límite de 4000 caracteres para la inserción de datos excepto la caja de texto del valor del modelo de madurez que tiene un límite de 1 carácter que debe ser numérico y estar entre 0 y 5 ambos incluidos.

Para cambiar el nombre o descripción de la información se debe seleccionar la tupla deseada del menú desplegable. Al hacer click sobre la tupla deseada, la caja de texto mostrará la información seleccionada para facilitar al administrador la tarea del cambio del nombre o descripción, para que en caso de que el administrador solo quisiera cambiar unos pocos caracteres en una tupla con mucha información, no tenga

que reescribir todo de nuevo. En cualquier caso se puede borrar todo por si se quiere escribir algo totalmente distinto. Para que funcione el efecto de que al pulsar sobre el menú desplegable, en la caja de texto aparezca el ítem del menú desplegable seleccionado, se necesita refrescar o actualizar la ventana para mostrar la información. Al realizar este refresco se puede observar el cambio pero en el caso de que se pulse un menú desplegable de la parte más baja de la ventana, es decir que se pulse un menú que esté por debajo de la barra de desplazamiento vertical, el efecto será que se actualizará correctamente pero el scroll vertical volverá hacia arriba en la ventana obligando a tener que volver a bajar para poder pulsar el botón aceptar y visualizar el cambio recién hecho, lo cual es bastante incómodo. Por ello en esta clase se ha utilizado una variable de tipo *Point*, más los métodos *View* de Java, que permite obtener el valor de las posiciones en la pantalla con el fin de que se pueda visualizar correctamente en cualquier punto en el que hagamos click y el scroll no se mueva al actualizar la pantalla impidiéndonos visualizar la información en el momento.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: "There are not X.", donde X pueden ser dominios, procesos, objetivos de control... cualquier información que no esté en la base de datos. Y si no hay información no aparecerá el botón de aceptar ni el menú desplegable ni tampoco la caja de texto para introducir datos, solamente una etiqueta de texto *JLabel* informando del mensaje.

El funcionamiento de los eventos cuando se pulsa el botón aceptar es el siguiente:

Se realiza un tratamiento de la cadena introducida o modificada de la original de la caja de texto. Una vez libre de espacios no útiles (los espacios al principio o final de la cadena no aportan ninguna información) y de caracteres problemáticos de inyección SQL se procede a realizar una serie de acciones sobre la base de datos antes de insertarla. Pero antes de las acciones en la base de datos, lo primero es comprobar si la cadena nueva es igual que la original, en caso de ser iguales se avisa y no se realiza ningún cambio. En caso de que no sean iguales, se comprueba en la base de datos si ya existe el mismo nombre, en cuyo caso se avisa de que ya existe y no se inserta ni se cambia nada. En caso de que el nombre sea distinto se realizan acciones de backup, borrado e inserción. El cambio de nombre de una tupla supone el borrado de la original tupla e insertado de la nueva tupla en la base de datos. Para llevar a cabo esto primero se realizan unas copias de seguridad (backup) de todas las tablas intermedias que estén relacionadas con la tupla que se quiere modificar. Los backups se realizan mediante el tipo de datos lista, en la lista se recoge toda la información de los atributos de las tuplas. Una vez se tienen los backups se procede al borrado de la tupla original, como el borrado es en cascada no es necesario borrar la tupla original de las tablas intermedias, ya que al borrar en cascada borrará la tupla original de la tabla propia y de las tablas intermedias que estén involucradas. Pero en algunas tuplas se ha hecho esto y en otras se borra a mano todas, pero el resultado es el mismo, la tupla original desaparece de las dos maneras, esto es así porque la base de datos cambió

durante el proceso de desarrollo. Una vez borrada la tupla original se inserta en su tabla la nueva tupla introducida en la caja de texto. Y a continuación con los backups que se hicieron en las listas obtenemos los atributos antiguos, generalmente el atributo código, para insertarlos con la nueva información en las tablas intermedias para que no se pierdan las relaciones entre tablas y la información quede intacta pero con un nuevo nombre o descripción. Después de hacer las operaciones sobre la base de datos se hace una llamada para crear la ventana de nuevo y realizar el refresco o actualización con la nueva información de las cajas y de los menús.

Si se pulsa el botón “Accept” sin elegir información del menú desplegable el sistema nos lo avisará con un mensaje de aviso.

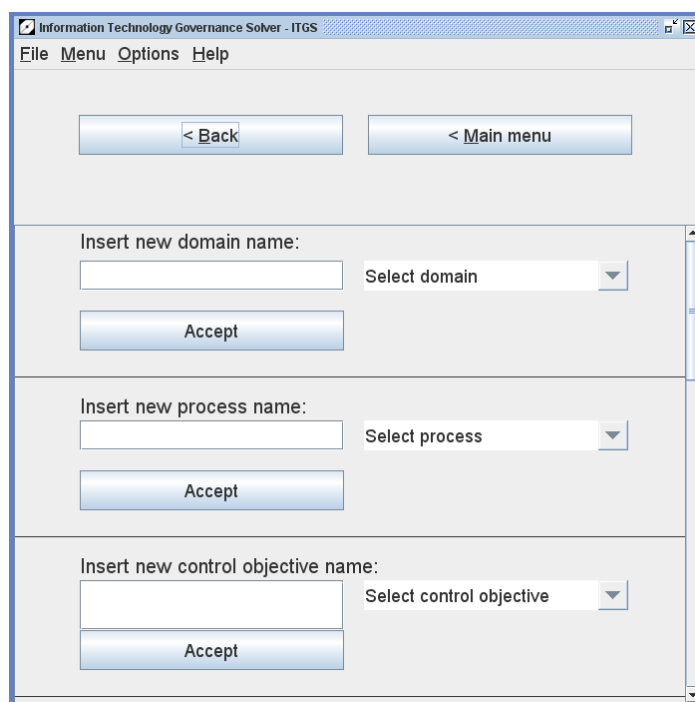


Ilustración 108. Cambio de nombres/descripciones en COBIT

El menú desplegable que se usa en cambiar nombre de objetivo de control y también en el de relacionar tablas que se va a explicar después, presenta un problema con la información del objetivo de control, que es el siguiente:

El objetivo de control en la base de datos contiene muchos caracteres y SQL de Oracle utilizado en la base de datos no acepta más de 2499 caracteres por línea por lo que se han tenido que usar saltos de línea y un procedimiento de inserción especial para insertar más caracteres cuando se superan los 4000 caracteres. Esto trae una serie de problemas, como por ejemplo que los menús desplegables no aceptan salto de línea ya que cada ítem que muestran es una sola línea por lo que al ponerlo todo en una línea, las líneas que tenían saltos salen juntas sin espacio.

Sin embargo, en la solución (tanto por pantalla como en fichero guardado) y en el área de texto de la izquierda al menú desplegable, cuando se usan estos datos de los menús salen correctamente con sus saltos ya que tienen otro tratamiento.

Para que aparecieran correctamente separados los caracteres que son de una línea, de la otra, en el menú desplegable se podría introducir un espacio más el carácter de retorno de carro (en código ASCII corresponde al valor 13, CHR (13)) después de cada salto de línea pero entonces, a la izquierda la información que saldría en el área de texto aparecería con un espacio de más, lo cual no tiene sentido. Otra solución habría sido realizar un tratamiento como el que se hace en la solución o en el guardado en el fichero, que es un tratado de palabra por palabra, pero habría complicado en exceso el código, sobre todo cuando se trata de un problema de la interpretación de los saltos de línea de Java. En la solución en cambio no notaríamos este efecto y no saldría con un espacio de más porque la solución ha sido tratada de diferente modo para poder mostrarla ajustada a la pantalla y en los ficheros que se guarden igual, también saldría correctamente.

En el apartado de cambiar nombres de objetivo de control ocurre un problema derivado de esto. Cuando pulsamos sobre un ítem del menú a la izquierda aparece de nuevo el texto pulsado en una caja de texto para ayudar al administrador, pero se ha tenido que usar un JTextArea con scroll (es un poco más grande que el resto de cajas para visualizar su propio scroll correctamente) en lugar de un JTextField para que interpretara bien los saltos de línea. El problema que encontramos es por la limitación de caracteres. Hay datos concretamente en 3 objetivos de control (relacionados con los procesos: PO4, PO10, DS4) que superan los 4000 caracteres y el límite del área de texto de introducción de datos es de 4000 caracteres. Por tanto si pinchamos sobre el menú no va a aparecer a la izquierda porque no cabe, se indica que no se puede introducir mediante un sonido “bip” por defecto de Java. Si aumentamos estos caracteres lógicamente si aparecerá al pulsar sobre el ítem. Pero no se puede aumentar ese tamaño ya que en el alta, si hacemos un alta con más de 4000 caracteres surgirá un error en la base de datos, que no permite con una instrucción típica de inserción introducir más de 4000 caracteres, habría que realizar un procedimiento especial de inserción y el sistema está diseñado para funcionar con inserciones normales de información sin procedimientos. Sin embargo, aunque no aparezca en el área de texto del lado izquierdo al pulsarlo, se puede cambiar el nombre si se desea, pero esta vez con un límite de 4000 caracteres.

En la cuestión del tratamiento de los caracteres hay que tener en cuenta lo siguiente. La cadena que se introduce recibe un tratamiento de caracteres para evitar la inserción de caracteres especiales y evitar el problema de la inyección SQL. Algunas tuplas de la base de datos insertadas con los scripts tienen algunos de estos caracteres como son la comilla simple, la comilla doble y la barra invertida. Estos caracteres son visualizados correctamente tanto en el menú como en las cajas, pero si deseamos introducir estos caracteres en las áreas de introducción de texto, la aplicación no lo va a permitir.

Si introducimos un cambio en el texto sobre el texto original con caracteres no permitidos y pulsamos el botón aceptar, lo que va a suceder es que la aplicación va a avisar de que la tupla ya existe en la base de datos. Esto ocurre porque a efectos del programa los caracteres no permitidos no existen, ya que no se tienen en cuenta, y por tanto aunque se cambien o añadan caracteres no permitidos, al no tratarse, no se compararán, sino que se comprobará la cadena sin ellos y por tanto la cadena nueva y la antigua serán la misma y se encontrará en la base de datos la misma tupla por lo que no se insertará la nueva. Esto ocurre cuando la cadena original no tiene ningún carácter no permitido en su script.

Si la cadena original tiene uno o varios caracteres no permitidos, e introducimos un cambio en el texto sobre el original con caracteres no permitidos también, lo que va a ocurrir es que el sistema va a detectar que hay un cambio y lo va a introducir en la base de datos. Como hemos dicho antes, esto no debería pasar ya que los caracteres nuevos no permitidos que se introducen son desechados porque no se tienen en cuenta, pero lo que sucede es que la cadena original tenía caracteres no permitidos también, que a su vez van a ser ignorados, y como consecuencia, la cadena nueva va a ser una réplica de la original pero sin sus caracteres no permitidos, por lo que va a ser distinta de la original y se insertará en la base de datos.

Si introducimos caracteres permitidos tendrán un cambio normal en la inserción.

En el caso del modelo de madurez, por su tratamiento distinto respecto al resto de información de COBIT, si aceptamos introduciendo espacios al principio o al final de la cadena nueva, el sistema no los tendrá en cuenta y avisará de que la cadena nueva coincide con la original. En el resto de información de COBIT, lo que diría el sistema es que ya existe una tupla en la base de datos con ese mismo nombre, no que sean iguales. Sin embargo ambos métodos son correctos.

El modelo de madurez y su valor tienen un tratamiento prácticamente igual que el de los anteriores de COBIT explicados pero con algunos controles más. El modelo de madurez está formado por el modelo de madurez y su valor. El valor debe ser numérico y debe estar entre 0 y 5 que son los valores que COBIT otorga en los modelos de madurez, donde 0 es Non-existent, 1 es Initial/Ad Hoc, 2 es Repeatable but Intuitive, 3 es Defined, 4 es Managed and Measurable y 5 es Optimised. Si se introduce un valor no válido el sistema nos lo indicará. En pantalla se muestra un signo interrogativo con el tooltip "Help" sobre él que nos explica estos valores numéricos mediante un panel de aviso si se pulsa sobre él.

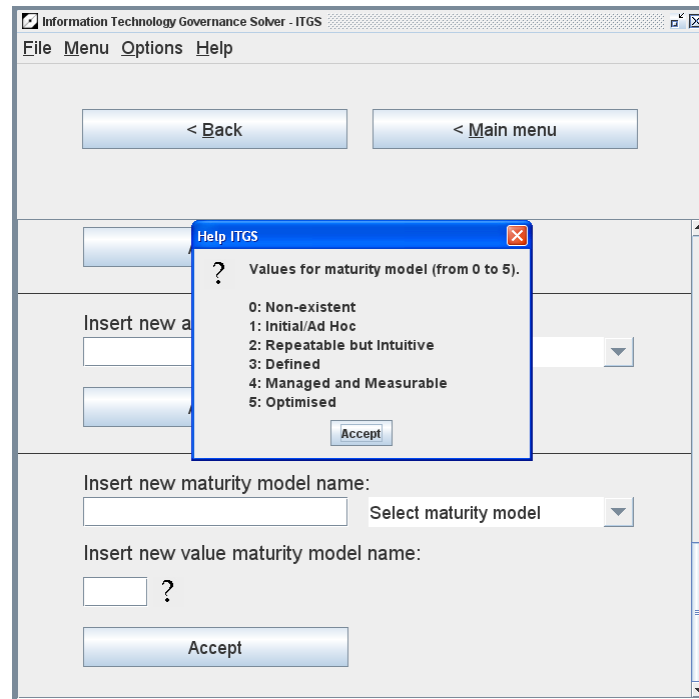


Ilustración 109. Modelo de madurez en COBIT

Se pueden realizar las siguientes acciones en modelo de madurez:

- Cambiar únicamente el nombre o descripción.
- Cambiar únicamente el valor.
- Cambiar el nombre o descripción y su valor.

Si se cambia solamente el nombre los pasos que se realizan son los de tratar la nueva cadena tal y como se explicó en los demás casos. Realizar un backup de las tablas intermedias y borrar los valores originales para después insertar con el nuevo nombre más el antiguo valor que tenía la tupla original.

Si se cambia solamente el valor se realiza lo mismo que lo anterior pero insertando el nombre o descripción del modelo de madurez del original con el nuevo valor, a no ser que el nuevo valor se esté usando ya en algún proceso, en cuyo caso se avisará de que no es posible hacer dicho cambio.

Si se cambian los dos atributos, nombre y valor, el procedimiento es tratar el texto introducido y realizar un backup de las tablas intermedias involucradas. Pero se comprueba también que el nuevo valor introducido no se esté usando ya en la tabla intermedia con proceso, ya que el modelo de madurez de COBIT va asociado siempre a un proceso, puesto que no tiene sentido que haya un valor repetido para un proceso. Es decir, un proceso debe tener un modelo de madurez asociado con valores de 0 a 5 pero no puede tener más de un valor para el 0, 1, 2, 3, 4 o 5. No puede haber un

modelo de madurez con varios valores del 0 al 5 para un proceso, debe tener como mucho un valor para cada valor del 0 al 5. Por lo tanto, si el nuevo valor ya está relacionado, el sistema no va a permitir que se introduzca y avisará con un mensaje al administrador. Si se cambia el atributo nombre y el valor es incorrecto porque se esté ya usando, el sistema introducirá el nuevo nombre pero con el valor antiguo, ignorando el erróneo ya en uso. Y en el caso de que el valor no se esté usando ya, lo cual es correcto, se realizan las tareas típicas como en el resto de casos, borrado de los valores originales para después insertar el nuevo nombre más el nuevo valor.

La llamada a la clase ***ventanaAdministrarCobitModificacionRelacion.java*** desde la ventana de modificación de Cobit crea la ventana que permite relacionar las tablas de la información en la base de datos de COBIT. En esta ventana existen dos menús desplegables para cada tabla que se puede relacionar en la base de datos y que a su vez está relacionada en COBIT. Las relaciones permitidas son las mismas relaciones existentes en la base de datos y son las siguientes:

- Dominio con Proceso
- Proceso con Objetivo de control
- Objetivo de control con Objetivo TI
- Objetivo TI con Objetivo de negocio
- Meta con Métrica
- Métrica con Objetivo de control
- Acción con Objetivo de control
- Modelo de madurez con Proceso

Cuando se crea la ventana toda esta información se obtiene desde la base de datos y se representa por pantalla en los menús desplegables.

Si algún conjunto de tuplas no contiene información se verá reflejado en la pantalla con el mensaje: "There are not X.", donde X pueden ser dominios, procesos, objetivos de control... cualquier información que no esté en la base de datos. Y si no hay información no aparecerá el botón de crear ni el de borrar relaciones, así como tampoco el menú desplegable correspondiente, solamente la etiqueta de texto JLabel informando del mensaje.

La ventana posee una barra de desplazamiento vertical para poder visualizar la ventana completamente y consta aparte de los menús desplegables, con sus propios scrolls vertical y horizontal en caso necesario, de dos botones para cada par de menús desplegables que permiten crear "Create relationship" o borrar "Delete relationship" una relación entre tablas.

Como se explicó anteriormente, la aplicación permite el insertado de información independiente y no restrictiva. Esto quiere decir que podemos introducir cualquier tipo de información pero se debe saber que en el alta la información no estará relacionada. Es por ello que en esta ventana de administración se permite

relacionar las tablas de toda la información que exista en la base de datos, en particular con todo lo relacionado con COBIT.

Para crear o borrar una relación se debe seleccionar un ítem de los dos menús desplegables que se deseen relacionar o desrelacionar. En caso de pulsar uno de los botones disponibles y no seleccionar los dos ítems el sistema lo indicará mediante un aviso.

En el caso de que se seleccionen ambos ítems de los menús y pulsemos el botón para crear la relación entre ellos, la aplicación comprueba en la base de datos si ambos ítems, que son tuplas, estaban ya relacionados o no. Esto se hace mediante una consulta a la tabla intermedia de ambas tuplas por atributo “código” que es clave primaria, si la consulta contiene valor es que están relacionadas, si no se devuelve nada en la consulta es que no hay relación. Si ambas tuplas estaban relacionadas y pulsamos el botón de crear relación, el sistema nos indicará que esas tuplas ya tenían relación y no hará nada. En cambio si la aplicación comprueba en la tabla intermedia de ambas tuplas que no existía relación, al pulsar en el botón de crear se creará la relación insertando los códigos de ambas tuplas en la tabla intermedia de ambas, creando de este modo la relación.

El proceso en el borrado de la relación es similar. Cuando se seleccionen ambas tuplas del par de menús desplegables se comprobará si existía o no una relación entre ellas mediante una consulta a la tabla intermedia. En caso de no existir relación entre las tuplas, al pulsar sobre el borrado de relación, la aplicación nos avisará de que no existe una relación previa y por tanto no hará nada. Pero si existía una relación anterior entre tuplas, la aplicación enviará a la base de datos una instrucción de borrado con los atributos código de ambas tablas de las dos tuplas para eliminar la tupla en la tabla intermedia, borrando de este modo la relación.

Como se puede observar, la tarea que se realiza en esta ventana, es la de inserción o borrado en la tabla intermedia que relaciona cada par de tablas de la base de datos.

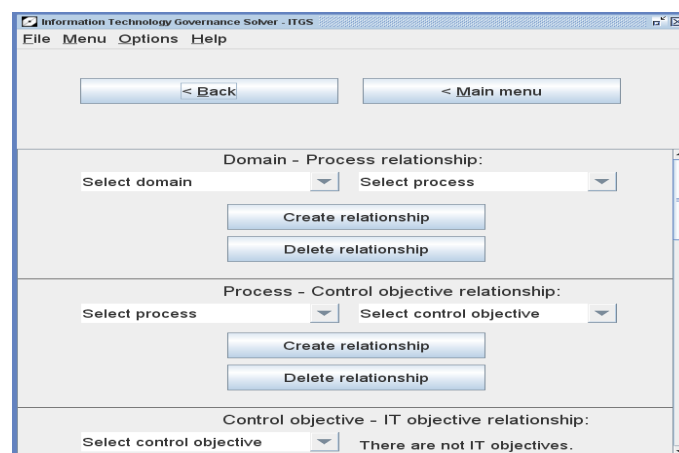


Ilustración 110. Creado y borrado de relación entre tablas de COBIT

La clase ***ventanaPreferenciasUsuario.java*** se crea cuando se pulsa el ítem *Preferences* desde el menú *Options* de la aplicación, en la ventana normal del programa, no en la ventana de la solución, cuyo menú como se ha explicado es distinto. Cuando se pulsa sobre el ítem se hace una llamada a esta clase, la cual crea una ventana en la que se pueden configurar una serie de aspectos de la aplicación a gusto del usuario. La ventana tiene en la parte superior un botón para volver a la ventana principal (“<Main menu”) y a continuación tres cajas de texto con botones asociados para cambiar información, un panel de color con un botón para aceptar el cambio y un botón más para reconfigurar la aplicación con los valores por defecto. Para visualizar esta ventana enteramente se dispone de una barra de desplazamiento vertical. En concreto los cambios que permite realizar esta ventana sobre el programa son los siguientes:

- Cambiar el número de problemas y de problemas relacionados que se muestran en la solución al usar las distintas ramas de consulta.
- Cambiar el número de evaluaciones que se muestran por ventana en la rama de consulta de las evaluaciones.
- Cambiar el tamaño de la letra de la barra del menú.
- Cambiar el color de la barra del menú.
- Cambiar a los valores por defecto de la aplicación.

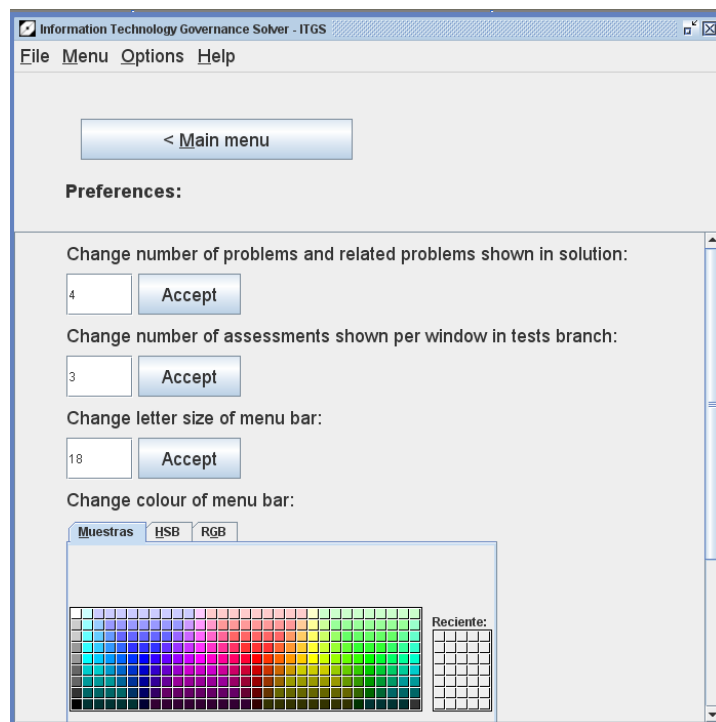


Ilustración 111. Vista de la parte superior de las Preferencias del usuario

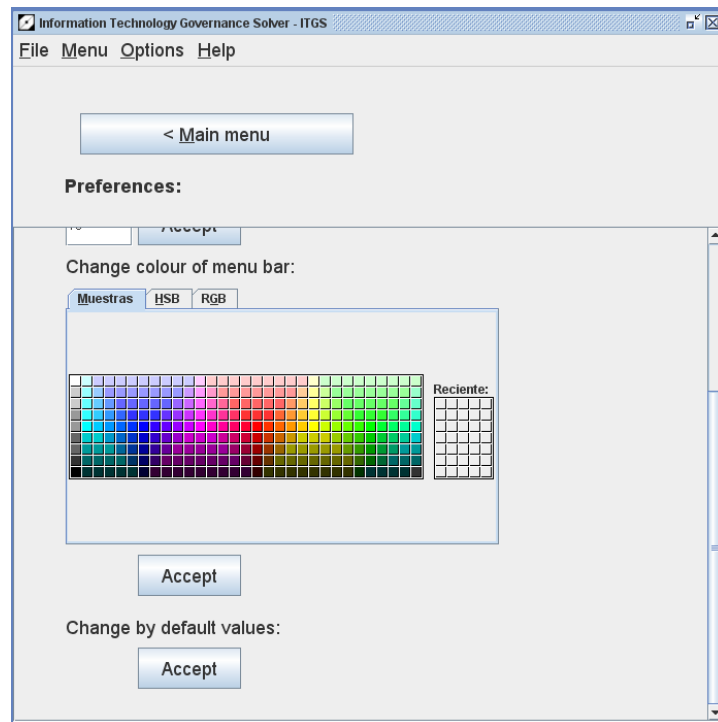


Ilustración 112. Vista de la parte inferior de las Preferencias del usuario

Cambiar el tamaño de la letra de la barra del menú:

La caja de texto para cambiar el tamaño de la letra de la barra del menú tiene un límite de 2 caracteres. Antes de cambiar su valor, se puede visualizar en la caja de texto el valor que tiene actualmente, es decir el valor del tamaño de letra actual que tiene el fichero que guarda la configuración de los parámetros (*ficheroConfiguracionParametrosSW.txt*). Cuando se pulsa sobre el botón asociado "Accept" lo primero que se comprueba es que el valor introducido sea un dígito, si es un carácter alfabético o no se introduce nada, se avisará de que se debe introducir un valor numérico. Si se introduce un valor numérico, este debe estar comprendido entre 10 y 40 puntos, ambos incluidos. Se ha elegido este rango porque un valor por debajo del 10 mostraría una letra difícil de leer, y por encima del 40 sería una letra de un tamaño bastante grande. Si el valor introducido está fuera del rango se muestra un mensaje avisando del rango correcto. Y si el valor está dentro del rango, se cambia el tamaño de la letra del menú, se muestra en la caja el nuevo valor y se guarda el nuevo valor en el fichero *ficheroConfiguracionParametrosSW.txt*.

Cambiar el color de la barra del menú:

El panel de color que viene implementado en Java por la clase *JColorChooser* permite seleccionar un color para la barra del menú. Cuando se pulsa sobre un color, automáticamente se produce el cambio visible en la barra de menú, pero es una

visualización momentánea ya que si cerramos y volvemos a ejecutar la aplicación, el color volverá a ser el que teníamos antiguamente. Para guardar el color elegido se debe pulsar el botón “Accept” que se encuentra debajo del panel y de esta manera, el código del color se guardará en el fichero *ficheroConfiguracionParametrosSW.txt* para que la próxima vez que se ejecute el programa pueda leer este código y se pueda visualizar el color elegido la siguiente vez.

Cambiar a los valores por defecto de la aplicación:

Si se pulsa el botón para cambiar los valores de la aplicación a los valores por defecto del programa, lo que hace el sistema es borrar el fichero *ficheroConfiguracionParametrosSW.txt* y crearlo de nuevo con los valores que por defecto la clase ***ficheroConfiguracionParametrosSW.java*** introduce desde la aplicación. A continuación, se destruye el *frame* de la aplicación y se crea uno nuevo comenzando desde la ventana principal, realizando una llamada al método *crearVentana* de la clase ***ventanaPrincipal.java***. Ha de realizarse esta nueva creación porque algunos cambios no podrían visualizarse en el momento y porque además, las ventanas que se crean a partir de las ventanas anteriores deben tener los mismos valores y por ello lo más óptimo es crear de nuevo la ventana principal del programa para que a partir de ella se creen las nuevas ventanas en concordancia con los mismos valores. Los valores que por defecto se aplican y que se encuentran en el fichero de configuraciones son:

4 -> número total de problemas encontrados que se muestran en la solución.

3 -> número de evaluaciones que se muestran por pantalla.

18 -> tamaño de la letra de la barra del menú.

English -> idioma inglés.

-1118482 -> código de decodificación del código RGB del color de la barra del menú que corresponde a un tipo de gris.

5.3.4 Ofuscación del código

Proteger el código fuente de cualquier aplicación es una tarea difícil y muchas veces es complicado conseguirlo al cien por cien.

En el caso concreto de Java, tras el compilado de las clases de la aplicación se genera por defecto un fichero ejecutable que tiene la extensión JAR. Este fichero ejecutable es el que se va a usar para ejecutar la aplicación y es el que, en el caso de una aplicación comercial, se utilizaría para su venta.

Pero el fichero JAR trae consigo un problema y es que a partir de él resulta muy fácil obtener el código fuente de una aplicación. Un fichero o archivo JAR está comprimido en formato ZIP con lo que si extraemos el contenido del archivo JAR mediante un descompresor de archivos ZIP se extraerán dos carpetas de la aplicación:

- La carpeta META-INF, que contiene el fichero llamado MANIFEST.MF que es un archivo que se usa para definir algunos datos como librerías usadas y donde se encuentra la clase principal de la aplicación entre otros.
- La carpeta con el mismo nombre del fichero JAR, cuyo contenido son todos los archivos con extensión CLASS compilados y todas las carpetas y ficheros necesarios para la ejecución del propio fichero JAR.

Con el contenido de estas carpetas, se puede realizar lo que se conoce como ingeniería inversa que consiste básicamente en descubrir como funciona un programa sin conocer su código fuente. Existen varios programas decompiladores de código, esto es, que hacen la operación inversa a la compilación o lo que es lo mismo traducir código de bajo nivel a más alto nivel, y de este modo realizar ingeniería inversa como el programa *DJ Java Decompiler*. Con este programa se puede obtener el código fuente de cada una de las clases que componen el fichero JAR y conocer así como está hecho el programa original antes de haber sido compilado para su posterior ejecución.

Cualquier aplicación Java puede ser víctima de ingeniería inversa y puede obtenerse su código fuente con lo que podría alterarse ya que sería comprensible fácilmente y por ende modificable, pudiéndose crear funcionamientos anómalos, no deseados... y poner en peligro todo el trabajo del creador original.

La solución a esto es hacer uso de lo que se conoce como ofuscación. La ofuscación es una técnica para conseguir un código menos legible por medio del cambio de nombres de variables por nombres sin sentido, añadido de funciones y cálculos innecesarios, creación de métodos con mucho código en lugar de crear métodos más pequeños...

Existen aplicaciones que realizan la ofuscación de código fuente pero puede realizarse a mano después o a medida que se desarrolla el código. Hay que tener

cuidado si se realiza después ya que algún mínimo cambio erróneo en la aplicación, por ejemplo una variable cambiada en un sitio y no en otro, podría llevar a funcionamientos anómalos del programa.

El código fuente de la aplicación se ha intentado hacer de una manera legible y con comentarios especialmente en las partes más complejas, pero se han utilizado el nombre de variables algo confusas en algunas clases en un intento de realizar técnicas de ofuscación.

5.3.5 Inyección SQL

La inyección SQL es un método de infiltración de código intruso que se aprovecha de una vulnerabilidad presente en el sistema informático de una aplicación en el momento de la validación de entrada para realizar consultas a una base de datos. Se conoce como inyección SQL al hecho de incrustar el código intruso y a la porción de código introducido dentro de otro.

La finalidad de la inyección SQL es que por medio del nuevo código incrustado se altere el funcionamiento normal de una aplicación y se logre ejecutar la porción de código inyectado, normalmente código que es de carácter malicioso, dañino o espía.

Esto quiere decir que en el momento en que la aplicación permite la introducción de datos por parte del usuario para que pueda hacer consultas a una base de datos, si este usuario malintencionado introdujera ciertos caracteres especiales y a continuación escribiera sentencias, estas interactuarían con la base de datos y podría destruir información o hacer consultas que no debería, etc.

Por ejemplo, imaginemos que el usuario puede introducir nombres para realizar consultas. Podría introducir el nombre 'Ana' en la casilla correspondiente de la aplicación e internamente el código ejecutaría la sentencia en la base de datos:

```
SELECT * FROM personas WHERE nombre='Ana';
```

Esta sentencia se ejecutaría correctamente y mostraría toda la información correspondiente a la persona cuyo nombre es Ana.

Pero si malintencionadamente el usuario introdujera en la casilla de la aplicación lo siguiente:

```
'Ana'; DROP TABLE personas;
```

Entonces la base de datos podría interpretar que el carácter ';' es el final de la sentencia y ejecutaría la siguiente sentencia 'DROP TABLE...' cuya ejecución en la base de datos borraría toda la tabla personas.

Por tanto existe un problema de seguridad cuando permitimos al usuario introducir datos en una aplicación que podría comprometer el sistema y hay que tratarlo para evitar esta situación.

El origen de esta vulnerabilidad está en el incorrecto tratamiento de las variables que un usuario puede introducir en una aplicación. Este problema está tratado en la aplicación de la siguiente manera:

Cuando un usuario puede introducir datos en la aplicación, no se hace una llamada instantánea a la base de datos sino que esos datos introducidos por el usuario son tratados y una vez hecho esto se hace la llamada a la base de datos con datos no problemáticos. Hay ciertos caracteres como las comillas, barras invertidas... (', /, ") que colocándolos en la introducción de datos podrían hacer que en la llamada a la base de datos entendiera mal la sentencia y si a continuación de esos caracteres se pone otra sentencia podría ejecutarla con el consiguiente riesgo. Por eso la cadena de caracteres que el usuario introduce está libre de estos caracteres problemáticos y solo se pueden introducir los caracteres que la aplicación tiene determinado con lo que se protege al programa de la inyección SQL.

El tratamiento que realiza la aplicación de la cadena de caracteres introducida por el usuario es el siguiente:

1. Se comprueba que la cadena introducida no sea todo espacios o esté vacía en cuyo caso se avisa con un mensaje al usuario indicando que no son datos válidos.
2. Si la cadena introducida es correcta se quitan todos los símbolos a excepción de letras, espacios en blanco, dígitos y ciertos caracteres especiales entre los que se encuentran: `° ¢ ¿ ? ¡ ! . , ; : () ~ [] - _` también son válidas las vocales que contengan tildes, tildes invertidas, acentos circunflejos y diéresis.
3. Posteriormente en un mayor refinamiento y consistencia se quitan los espacios que puedan encontrarse al inicio o al final de la cadena ya que no aportan ningún tipo de información.

Con este tratamiento de caracteres se puede realizar la llamada a la base de datos con la seguridad de que no se producirá una inyección SQL.

Existen otras maneras de evitar la inyección SQL, por ejemplo usando una clase especial de JDBC, la clase *PreparedStatement*, pero en la aplicación se ha optado por hacer un tratamiento de las cadenas de caracteres introducidas por el usuario.

5.3.6 Scripts de la base de datos

Los scripts o archivos de órdenes son los ficheros de texto, con extensión SQL, que se han utilizado para introducir, de una sola vez, toda la información que la aplicación va a usar desde la base de datos. Hay dos scripts, uno para definir y crear todas las tablas llamado *script_tablas.sql* y otro para insertar todos los datos sobre las tablas llamado *script_insert.sql*.

Al introducir el script de las tablas, la primera operación que se realiza es el borrado de todas las tablas existentes para que no haya error al crear las tablas. Las tablas constan todas de un atributo código, que es la clave primaria, la clave identificativa, y de otros atributos descriptivos. Las tablas *palabra* y *sinonimo* no tienen el atributo código ya que su propia descripción es única e identificativa y por tanto su atributo de descripción es su clave primaria. Las tablas intermedias están formadas por los atributos que sean clave primaria de las tablas relacionadas y con clave foránea referencian a sus tablas. El total de tablas que se introducen es de 25 y es el siguiente:

Tablas:

accion, meta, metrica, objetivo_de_control, objetivo_ti, objetivo_de_negocio, proceso, modelo_de_madurez, evaluacion, dominio, problema, sinonimo, palabra.

Tablas intermedias (relaciones entre tablas):

tiene_asociada, mide, se_evalua, tiene_asociado, contiene, se_asocia, se_relaciona, evalua, procesa, domina, pertenece, tiene.

El borrado de las claves foráneas se ha definido como ON DELETE CASCADE para que en el caso de un borrado se borre la información en las tablas intermedias.

5.3.7 Instalación y ejecución de la aplicación

La instalación de la aplicación IT Governance Solver v. 1.0 se realiza de forma muy sencilla, y consta de dos pasos, que deben ser realizados una vez que el sistema dispone de los requisitos de funcionamiento:

Descompresión del contenido del fichero comprimido *IT Governance Solver v. 1.0.zip* a una ubicación cualquiera desde donde se quiera ejecutar el programa. Se recomienda descomprimir el fichero en el Escritorio de Windows, a fin de que sea más sencilla su ejecución. Dicho fichero comprimido contiene un acceso directo llamado *IT Governance Solver v. 1.0*, una carpeta llamada *IT Governance Solver v. 1.0*, y por último otra carpeta con el nombre Scripts.

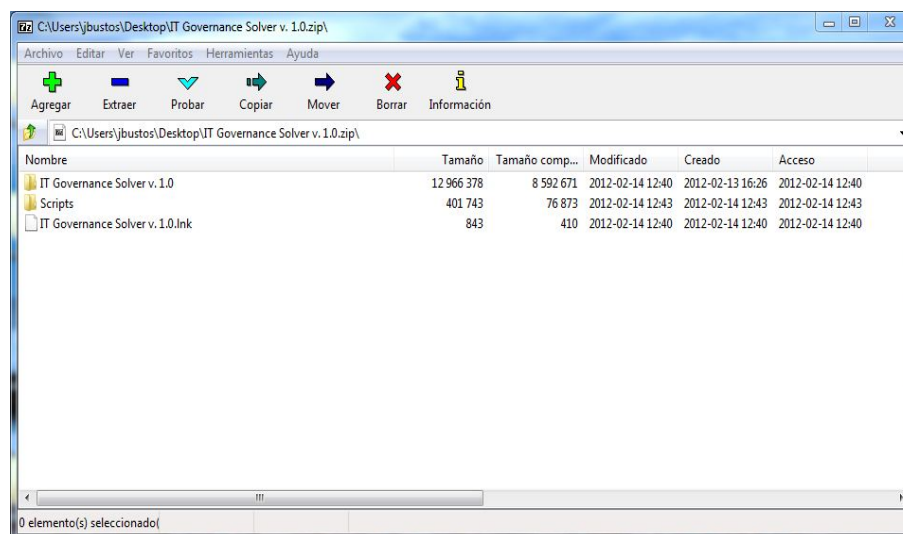


Ilustración 113. Contenido del fichero comprimido

Una vez descomprimido todo el contenido, hay que abrir la carpeta Scripts, e introducir en la base de datos del sistema –por ejemplo mediante SQL Plus–, los scripts de la base de datos de la siguiente forma:

Primero se introduce el script *script_tablas.sql*, que es el encargado de borrar las tablas del programa en la base de datos en el caso de que ya existieran, y de crearlas para posteriormente introducir los datos iniciales del programa.

Por último, se introduce el *script_insert.sql*, que contiene todos los datos iniciales del programa como son los problemas con sus palabras y sinónimos, las evaluaciones, y la metodología COBIT 4.1.

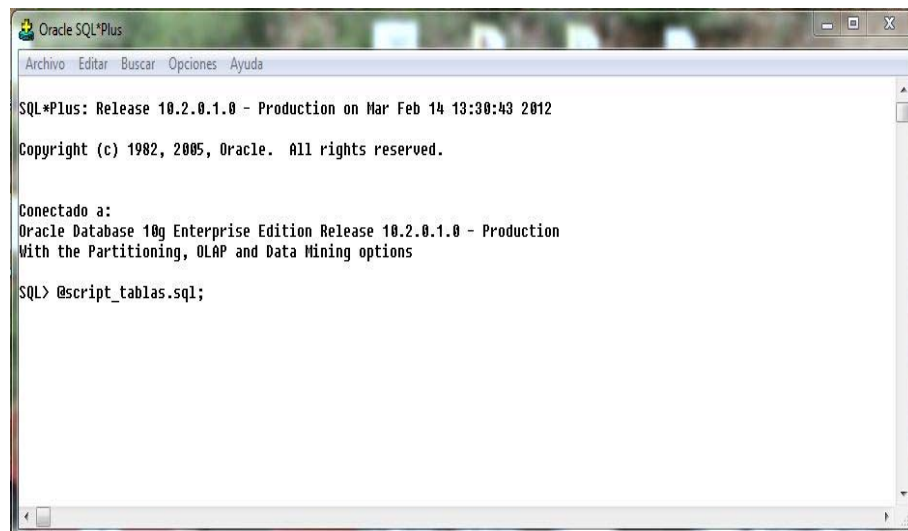


Ilustración 114. Inserción de los scripts en la base de datos

Una vez realizados estos pasos el programa queda totalmente instalado en el sistema.

6. Pruebas del Sistema

6.1 Objetivo

El objetivo de este apartado es asegurar y verificar el correcto funcionamiento de la aplicación, mediante una serie de pruebas que tienen como objetivo cubrir el mayor número de situaciones de error, riesgos o excepciones encontrados en el entorno de pruebas definido.

Lo que se busca es corregir el máximo número posible de situaciones de error, así como minimizar el mayor número de defectos de codificación posible.

6.2 Especificación del entorno de pruebas

En este punto se describe de forma detallada el entorno necesario para la realización de las pruebas del sistema. Dicho entorno de pruebas, se creará con la intención de simular hasta el más mínimo detalle del entorno operativo, para así poder garantizar que el sistema funcionará en las condiciones reales de uso. Para ello se cuenta con la siguiente configuración:

REQUISITOS HARDWARE:

Procesador Intel Pentium i7 2,66 GHz

4 GB RAM

2 discos SATA de 1 TB conectados en RAID 1

Lector DVD

Conexión a Internet de 30 Mbits/s

Impresora multifunción HP Photosmart 5510

REQUISITOS SOFTWARE:

Microsoft Windows 7 Professional (x64)

Base de Datos Oracle 11g

Java SE 7

Compresor/descompresor 7-zip

Adobe Reader X

Especificación técnica de los niveles de prueba

En este apartado se describirán las pruebas a realizar sobre el sistema, con el objetivo de verificar su correcto funcionamiento.

6.3 Niveles de pruebas a emplear

Tomando el elemento a probar como si fuese una caja que recibe entradas, que realiza un proceso y produce una salida, se pueden clasificar las pruebas en dos tipos:

Caja negra: este tipo de pruebas, se centran en verificar los requisitos funcionales del sistema, comprobando que cumplen todas sus funciones. Las pruebas se aplican sobre el sistema empleando un conjunto de datos de entrada y observando las salidas que se producen, para determinar si la función se está realizando correctamente.

Caja blanca: tienen por objetivo verificar el funcionamiento de las estructuras de control del código del sistema, probando para ello, las decisiones lógicas en sus vertientes, verdadero y falso.

6.4 Plantilla de definición de pruebas

Para conseguir una definición de las pruebas completa y consistente, se define la plantilla de definición de pruebas:

Identificador	
Descripción	
Paquete del sistema	
Objetivo	
Tipo de procedimiento	
Procedimiento	
Precondiciones	
Poscondiciones	
Nivel de aseguramiento	
Resultado	<input type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 72. Plantilla para la definición de las Pruebas

A continuación se procede a realizar una breve descripción de los campos que conforman la plantilla:

Identificador: La nomenclatura seguida para identificar unívocamente a los componentes descritos será: **PR-TIPO-XXX**

PR: Siglas de la palabra Pruebas.

TIPO: Tipo de prueba que se realiza. Si es de caja negra (NE) o de caja blanca (BL).

XXX: Indica el número de prueba, siendo un valor único, empezando desde el 001.

Descripción: Nombre de la prueba.

Paquete del sistema: Nombre del paquete del sistema al que corresponde la prueba.

Objetivo: Propósito de la prueba.

Tipo de procedimiento: Especificar si se trata de un procedimiento de “caja blanca” o “caja negra”.

Procedimiento: Pasos a seguir para realizar la prueba.

Precondiciones: Condiciones que el sistema debe cumplir antes de iniciar la prueba.

Poscondiciones: Estado en el que se debe quedar el sistema después de ejecutar la prueba, es decir, el resultado esperado.

Nivel de aseguramiento: Porcentaje de código del componente que se está probando. Nivel de calidad que la indica en términos de completitud sobre el elemento que se está probando.

Resultado: Indica si la prueba se realizó de manera satisfactoria o no.

6.5 Pruebas de caja negra

PR-NE-001	
Descripción	Prueba para el paquete del sistema <i>Sistema Principal</i>
Paquete del sistema	<i>Sistema Principal</i>
Objetivo	Poner en funcionamiento el sistema desde el fichero ejecutable
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: No tiene • Parámetro de salida: int resultado
Precondiciones	No procede
Poscondiciones	Sistema iniciado
Nivel de aseguramiento	25%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 73. PR-NE-001

PR-NE-002	
Descripción	Prueba para el paquete del sistema <i>Parámetros_SW</i>
Paquete del sistema	<i>Parámetros_SW</i>
Objetivo	Cambiar la configuración de los parámetros de la interfaz de la aplicación
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Parámetros de la interfaz modificados
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 74. PR-NE-002

PR-NE-003	
Descripción	Prueba para el paquete del sistema <i>Conexión</i>
Paquete del sistema	<i>Conexión</i>
Objetivo	Comprobar la conectividad de la aplicación con la Base de Datos de Oracle
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Sistema conectado con la Base de Datos de Oracle
Nivel de aseguramiento	30%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 75. PR-NE-003

PR-NE-004	
Descripción	Prueba para el paquete del sistema <i>Ficheros</i>
Paquete del sistema	<i>Ficheros</i>
Objetivo	Comprobar la creación en el caso de no existir, y la lectura en el caso de existir, de los ficheros de texto plano utilizados por la aplicación
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Ficheros creados/leídos correctamente
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 76. PR-NE-004

PR-NE-005	
Descripción	Prueba para el paquete del sistema <i>Usuario</i>
Paquete del sistema	<i>Usuario</i>
Objetivo	Mostrar la ventana del menú principal del usuario
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int • Parámetro de salida: int resultado
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Ventana del menú principal del usuario mostrada
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 77. PR-NE-005

PR-NE-006	
Descripción	Prueba para el paquete del sistema <i>Usuario_COBIT</i>
Paquete del sistema	<i>Usuario_COBIT</i>
Objetivo	Gestionar la selección de dominios y procesos de COBIT 4.1 por parte del usuario
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int • Parámetro de salida: int resultado
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Selección de dominios y procesos de COBIT 4.1 realizada correctamente
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 78. PR-NE-006

PR-NE-007	
Descripción	Prueba para el paquete del sistema <i>Solución</i>
Paquete del sistema	<i>Solución</i>
Objetivo	Gestionar la solución de la aplicación
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado y string solucion
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Solución mostrada correctamente
Nivel de aseguramiento	50%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 79. PR-NE-007

PR-NE-008	
Descripción	Prueba para el paquete del sistema <i>Administrador</i>
Paquete del sistema	<i>Administrador</i>
Objetivo	Controlar el acceso al menú principal de administración del administrador, por medio de una contraseña
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int • Parámetro de salida: int resultado y string solucion
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Menú principal del administrador mostrado correctamente
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 80. PR-NE-008

PR-NE-009	
Descripción	Prueba para el paquete del sistema <i>Administrador_BBDD</i>
Paquete del sistema	<i>Administrador_BBDD</i>
Objetivo	Gestionar la configuración de los parámetros de conexión con la Base de Datos, comprobando la conectividad con la misma
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado y string resultado
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación de los parámetros de conexión con la Base de Datos, y conectividad comprobada
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 81. PR-NE-009

PR-NE-010	
Descripción	Prueba para el paquete del sistema <i>Administrador_Contraseña</i>
Paquete del sistema	<i>Administrador_Contraseña</i>
Objetivo	Gestionar el cambio de contraseña del administrador
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado y string resultado2
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación de la contraseña del administrador
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 82. PR-NE-010

PR-NE-011	
Descripción	Prueba para el paquete del sistema <i>Administrador_Problema</i>
Paquete del sistema	<i>Administrador_Problema</i>
Objetivo	Gestionar el cambio de nombres y las relaciones de la rama de consulta de problemas del usuario
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado y string resultado2
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación correcta de los parámetros elegidos por el administrador
Nivel de aseguramiento	30%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 83. PR-NE-011

PR-NE-012	
Descripción	Prueba para el paquete del sistema <i>Administrador_Evaluación</i>
Paquete del sistema	<i>Administrador_Evaluación</i>
Objetivo	Gestionar el cambio de nombres y las relaciones de la rama de tests de evaluación del usuario
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado y string resultado2
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación correcta de los parámetros elegidos por el administrador
Nivel de aseguramiento	40%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 84. PR-NE-012

PR-NE-013	
Descripción	Prueba para el paquete del sistema <i>Administrador_COBIT</i>
Paquete del sistema	<i>Administrador_COBIT</i>
Objetivo	Gestionar el cambio de nombres y las relaciones de la rama de COBIT 4.1 del usuario
Tipo de procedimiento	Caja Negra
Procedimiento	<ul style="list-style-type: none"> • Parámetros de entrada: int y string • Parámetro de salida: int resultado y string resultado2
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación correcta de los parámetros elegidos por el administrador
Nivel de aseguramiento	50%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 85. PR-NE-013

6.6 Pruebas de caja blanca

PR-BL-001	
Descripción	Prueba para el paquete del sistema <i>Sistema Principal</i>
Paquete del sistema	<i>Sistema Principal</i>
Objetivo	Poner en funcionamiento el sistema desde el fichero ejecutable
Tipo de procedimiento	Caja Blanca
Procedimiento	Iniciar la aplicación
Precondiciones	No tiene
Poscondiciones	Sistema iniciado correctamente
Nivel de aseguramiento	25%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 86. PR-BL-001

PR-BL-002	
Descripción	Prueba para el paquete del sistema <i>Parámetros_SW</i>
Paquete del sistema	<i>Parámetros_SW</i>
Objetivo	Cambiar la configuración de los parámetros de la interfaz de la aplicación
Tipo de procedimiento	Caja Blanca
Procedimiento	Ejecutar la ventana de opciones de la interfaz
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Parámetros de la interfaz modificados
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 87. PR-BL-002

PR-BL-003	
Descripción	Prueba para el paquete del sistema <i>Conexión</i>
Paquete del sistema	<i>Conexión</i>
Objetivo	Comprobar la conectividad de la aplicación con la Base de Datos de Oracle
Tipo de procedimiento	Caja Blanca
Procedimiento	Ejecutar la rama del usuario de la aplicación. Mostrar el menú principal del usuario
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Sistema conectado con la Base de Datos de Oracle
Nivel de aseguramiento	30%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 88. BL-NE-003

PR-BL-004	
Descripción	Prueba para el paquete del sistema <i>Ficheros</i>
Paquete del sistema	<i>Ficheros</i>
Objetivo	Comprobar la creación en el caso de no existir, y la lectura en el caso de existir, de los ficheros de texto plano utilizados por la aplicación
Tipo de procedimiento	Caja Blanca
Procedimiento	Eliminar los tres ficheros de configuración y ejecutar el programa. Ver si tras ejecutar el programa están creados de nuevo. Después, leer la contraseña del fichero que la contiene, así como los parámetros modificables de la interfaz, y los parámetros de conexión con la Base de Datos
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Ficheros creados/leídos correctamente
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 89. PR-BL-004

PR-BL-005	
Descripción	Prueba para el paquete del sistema <i>Usuario</i>
Paquete del sistema	<i>Usuario</i>
Objetivo	Mostrar la ventana del menú principal del usuario
Tipo de procedimiento	Caja Blanca
Procedimiento	Ejecutar la aplicación e ir navegando hasta el menú principal del usuario
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Ventana del menú principal del usuario mostrada
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 90. PR-BL-005

PR-BL-006	
Descripción	Prueba para el paquete del sistema <i>Usuario_COBIT</i>
Paquete del sistema	<i>Usuario_COBIT</i>
Objetivo	Gestionar la selección de dominios y procesos de COBIT 4.1 por parte del usuario
Tipo de procedimiento	Caja Blanca
Procedimiento	Selección de dominios y procesos de COBIT 4.1
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Selección de dominios y procesos de COBIT 4.1 realizada correctamente
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 91. PR-BL-006

PR-BL-007	
Descripción	Prueba para el paquete del sistema <i>Solución</i>
Paquete del sistema	<i>Solución</i>
Objetivo	Gestionar la solución de la aplicación
Tipo de procedimiento	Caja Blanca
Procedimiento	Mostrar la solución por pantalla a través de cualquiera de las tres ramas por las que puede hacer una consulta el usuario. Además, imprimir la primera hola de la solución a modo de demostración, y almacenarla en los formatos txt, PDF, y HTML
Precondiciones	Sistema iniciado correctamente
Poscondiciones	Solución mostrada correctamente
Nivel de aseguramiento	50%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 92. PR-BL-007

PR-BL-008	
Descripción	Prueba para el paquete del sistema <i>Administrador</i>
Paquete del sistema	<i>Administrador</i>
Objetivo	Controlar el acceso al menú principal de administración del administrador, por medio de una contraseña
Tipo de procedimiento	Caja Blanca
Procedimiento	Navegar hasta la introducción de la contraseña de administrador, e introducirla correctamente
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Menú principal del administrador mostrado correctamente
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 93. PR-BL-008

PR-BL-009	
Descripción	Prueba para el paquete del sistema <i>Administrador_BBDD</i>
Paquete del sistema	<i>Administrador_BBDD</i>
Objetivo	Gestionar la configuración de los parámetros de conexión con la Base de Datos, comprobando la conectividad con la misma
Tipo de procedimiento	Caja Blanca
Procedimiento	Navegar hasta la modificación de los parámetros de la conexión con la Base de Datos, e introducir estos de forma correcta. Comprobar que la conectividad se realiza de forma satisfactoria con la Base de Datos
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación de los parámetros de conexión con la Base de Datos, y conectividad comprobada
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 94. PR-BL-009

PR-BL-010	
Descripción	Prueba para el paquete del sistema <i>Administrador_Contraseña</i>
Paquete del sistema	<i>Administrador_Contraseña</i>
Objetivo	Gestionar el cambio de contraseña del administrador
Tipo de procedimiento	Caja Blanca
Procedimiento	Navegar hasta la ventana de cambio de contraseña de administrador, y realizar el cambio de la misma
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación de la contraseña del administrador
Nivel de aseguramiento	10%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 95. PR-BL-010

PR-BL-011	
Descripción	Prueba para el paquete del sistema <i>Administrador_Problema</i>
Paquete del sistema	<i>Administrador_Problema</i>
Objetivo	Gestionar el cambio de nombres y las relaciones de la rama de consulta de problemas del usuario
Tipo de procedimiento	Caja Blanca
Procedimiento	Navegar hasta las pantallas de cambio de nombres y de relaciones de la rama de consulta de problemas del usuario, y realizar cambios
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación correcta de los parámetros elegidos por el administrador
Nivel de aseguramiento	30%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 96. PR-BL-011

PR-BL-012	
Descripción	Prueba para el paquete del sistema <i>Administrador_Evaluación</i>
Paquete del sistema	<i>Administrador_Evaluación</i>
Objetivo	Gestionar el cambio de nombres y las relaciones de la rama de tests de evaluación del usuario
Tipo de procedimiento	Caja Blanca
Procedimiento	Navegar hasta las pantallas de cambio de nombres y de relaciones de la rama de tests de evaluación del usuario, y realizar cambios
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación correcta de los parámetros elegidos por el administrador
Nivel de aseguramiento	30%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 97. PR-BL-012

PR-BL-013	
Descripción	Prueba para el paquete del sistema <i>Administrador_COBIT</i>
Paquete del sistema	<i>Administrador_COBIT</i>
Objetivo	Gestionar el cambio de nombres y las relaciones de la rama de COBIT 4.1 del usuario
Tipo de procedimiento	Caja Blanca
Procedimiento	Navegar hasta las pantallas de cambio de nombres y de relaciones de la rama de COBIT 4.1 del usuario, y realizar cambios
Precondiciones	Sistema iniciado correctamente y contraseña correctamente introducida
Poscondiciones	Modificación correcta de los parámetros elegidos por el administrador
Nivel de aseguramiento	50%
Resultado	<input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto

Tabla 98. PR-BL-013

7. Análisis Económico

7.1 Estimación inicial mediante COCOMO II 2000

7.1.1 Introducción

Para estimar el tamaño de la aplicación y del esfuerzo para desarrollarla, se ha utilizado el software COCOMO II 2000.

El modelo COCOMO, antecesor de COCOMO II, fue presentado por Barry Boehm en 1981 y se convirtió en el más conocido y referenciado, además del más documentado de todos los modelos de estimación de esfuerzo de las actividades de diseño, codificación, pruebas y mantenimiento.

La versión inicial de COCOMO se obtuvo a partir de la revisión de los modelos de costes existentes, en la cual participaron varios expertos en dirección de proyectos, los cuales poseían además cierta experiencia en la utilización de diferentes modelos de estimación.

En los 90, las técnicas de desarrollo de software cambiaron dramáticamente, surgió la necesidad de reusar software existente, la construcción de sistemas usando librerías, etc. Estos cambios comenzaron a generar problemas en la aplicación del modelo COCOMO. La solución fue reinventar el modelo. Después de algunos años y de un esfuerzo combinado de USC-CSE (University of Southern California- Center For Software Engineering), IRUS at UC Irvine y organizaciones privadas, aparece COCOMO II.

COCOMO II consta de tres modelos, cada uno de los cuales ofrece una precisión acorde a cada etapa de desarrollo del proyecto. Enunciados en orden creciente de fidelidad son, modelo de Composición de Aplicación, Diseño Temprano y Post Arquitectura.

7.1.2 Grupos de Datos Internos (ILF)

- Almacén PALABRA:

RET	DET
PALABRA	PALABRA
1	1
Complejidad baja	

Tabla 99. Almacén PALABRA

- Almacén SINONIMO:

RET	DET
SINONIMO	SINONIMO
1	1
Complejidad baja	

Tabla 100. Almacén PALABRA

- Almacén TIENE:

RET	DET
TIENE	PALABRA
	SINONIMO
1	2
Complejidad baja	

Tabla 101. Almacén TIENE

- **Almacén PROBLEMA:**

RET	DET
PROBLEMA	COD_PROBLEMA
	PROBLEMA
1	2
Complejidad baja	

Tabla 102. Almacén PROBLEMA

- **Almacén PERTENECE:**

RET	DET
PERTENECE	PALABRA
	COD_PROBLEMA
1	2
Complejidad baja	

Tabla 103. Almacén PERTENECE

- **Almacén PROCESO:**

RET	DET
PROCESO	COD_PROCESO
	DESCRIPCION
1	2
Complejidad baja	

Tabla 104. Almacén PROCESO

- **Almacén PROCESA:**

RET	DET
PROCESA	COD_PROBLEMA
	COD_PROCESO
1	2
Complejidad baja	

Tabla 105. Almacén PROCESA

- **Almacén EVALUACION:**

RET	DET
EVALUACION	COD_EVALUACION PREGUNTA
1	2
Complejidad baja	

Tabla 106. Almacén EVALUACION

- **Almacén EVALUA:**

RET	DET
EVALUA	COD_EVALUACION COD_PROCESO
1	2
Complejidad baja	

Tabla 107. Almacén EVALUA

- **Almacén DOMINIO:**

RET	DET
DOMINIO	COD_DOMINIO DESCRIPCION
1	2
Complejidad baja	

Tabla 108. Almacén DOMINIO

- **Almacén DOMINA:**

RET	DET
DOMINA	COD_DOMINIO COD_PROCESO
1	2
Complejidad baja	

Tabla 109. Almacén DOMINA

- Almacén MODELO_DE_MADUREZ:

RET	DET
MODELO_DE_MADUREZ	COD_MODELO_MADUREZ VALOR_PREGUNTA PREGUNTA
1	3
Complejidad baja	

Tabla 110. Almacén MODELO_DE_MADUREZ

- Almacén SE_RELACIONA:

RET	DET
SE_RELACIONA	COD_MODELO_MADUREZ COD_PROCESO
1	2
Complejidad baja	

Tabla 111. Almacén SE_RELACIONA

- Almacén OBJETIVO_DE_CONTROL:

RET	DET
OBJETIVO_DE_CONTROL	COD_OBJETIVO_CONTROL DESCRIPCION
1	2
Complejidad baja	

Tabla 112. Almacén OBJETIVO_DE_CONTROL

- **Almacén CONTIENE:**

RET	DET
CONTIENE	COD_PROCESO
	COD_OBJETIVO_CONTROL
1	2
Complejidad baja	

Tabla 113. Almacén CONTIENE

- **Almacén ACCION:**

RET	DET
ACCION	COD_ACCION
	COD_DESCRIPCION
1	2
Complejidad baja	

Tabla 114. Almacén ACCION

- **Almacén TIENE_ASOCIADA:**

RET	DET
TIENE_ASOCIADA	COD_OBJETIVO_CONTROL
	COD_ACCION
1	2
Complejidad baja	

Tabla 115. Almacén TIENE_ASOCIADA

- **Almacén METRICA:**

RET	DET
METRICA	COD_METRICA
	DESCRIPCION
1	2
Complejidad baja	

Tabla 116. Almacén METRICA

- **Almacén META:**

RET	DET
META	COD_META COD_DESCRIPCION
1	2
Complejidad baja	

Tabla 117. Almacén META

- **Almacén MIDE:**

RET	DET
MIDE	COD_METRICA COD_META
1	2
Complejidad baja	

Tabla 118. Almacén MIDE

- **Almacén SE_EVALUA:**

RET	DET
SE_EVALUA	COD_OBJETIVO_CONTROL COD_METRICA
1	2
Complejidad baja	

Tabla 119. Almacén SE_EVALUA

- **Almacén OBJETIVO_TI:**

RET	DET
OBJETIVO_TI	COD_OBJETIVO_TI DESCRIPCION
1	2
Complejidad baja	

Tabla 120. Almacén OBJETIVO_TI

- Almacén OBJETIVO_DE_NEGOCIO:

RET	DET
OBJETIVO_DE_NEGOCIO	COD_OBJETIVO_NEGOCIO DESCRIPCION
1	2
Complejidad baja	

Tabla 121. Almacén OBJETIVO_DE_NEGOCIO

- Almacén SE_ASOCIA:

RET	DET
SE_ASOCIA	COD_OBJETIVO_NEGOCIO COD_OBJETIVO_TI
1	2
Complejidad baja	

Tabla 122. Almacén SE_ASOCIA

- Almacén TIENE_ASOCIADO:

RET	DET
TIENE_ASOCIADO	COD_OBJETIVO_TI COD_OBJETIVO_CONTROL
1	2
Complejidad baja	

Tabla 123. Almacén SE_ASOCIADO

- Almacén CONTRASEÑA_ADMINISTRADOR

RET	DET
CONTRASEÑA_ADMINISTRADOR	CONTRASEÑA
1	1
Complejidad baja	

Tabla 124. Almacén CONTRASEÑA_ADMINISTRADOR

- Almacén CONFIGURACION_BASE_DE_DATOS:

RET	DET
CONFIGURACION_BASE_DE_DATOS	USUARIO
	CONTRASEÑA
	NOMBRE_BBDD
	PUERTO
	HOST
1	5
Complejidad baja	

Tabla 125. Almacén CONFIGURACION_BASE_DE_DATOS

- Almacén PARAMETROS_INTERFAZ:

RET	DET
PARAMETROS_INTERFAZ	IDIOMA
	TAMAÑO_LETRA
	COLOR_BARRA_SUPERIOR
	NUMERO_PROBLEMAS
	NUMERO_TESTS_EVALUACION
1	5
Complejidad baja	

Tabla 126. Almacén PARAMETROS_INTERFAZ

7.1.3 Grupos de Datos Externos (ELF)

No existen Grupos de Datos Externos, ya que no existe un grupo de datos relacionados, identificables por el usuario o información de control utilizada por la aplicación, pero mantenida por otro sistema.

7.1.4 Entradas Externas (EI)

ENTRADA	FTR	DET	COMPLEJIDAD
consultaProblemas	6	1	BAJA
realizarTestsEvaluacion	5	1	BAJA
consultaCOBIT	6	1	BAJA
modificarPreferenciasInterfaz	1	1	BAJA
consultaAyuda	1	1	BAJA
modificarContraseñaAdministrador	1	1	BAJA
modificarParametrosBaseDatos	1	1	BAJA
modificarInformacionProblemas	2	3	MEDIA
modificarInformacionTestsEvaluacion	2	3	MEDIA
modificarInformacionCOBIT	3	5	MEDIA

Tabla 127. Entradas Externas de Usuario

7.1.5 Salidas Externas (EO)

SALIDA	FTR	DET	COMPLEJIDAD
mensaje_ok_consulta_problemas	1	1	BAJA
mensaje_ok_consulta_tests_evaluacion	1	1	BAJA
mensaje_ok_consulta_COBIT	1	1	BAJA
mensaje_ok_cambiar_preferencias_interfaz	1	1	BAJA
mensaje_error_cambiar_preferencias_interfaz	1	1	BAJA
mensaje_ok_cambiar_contraseña_administrador	1	1	BAJA
mensaje_error_cambiar_contraseña_administrador	1	1	BAJA
mensaje_ok_cambiar_parametros_BaseDatos	1	1	BAJA
mensaje_error_cambiar_parametros_BaseDatos	1	1	BAJA
mensaje_ok_modificar_informacion_problemas	1	1	BAJA
mensaje_error_modificar_informacion_problemas	1	1	BAJA
mensaje_ok_modificar_informacion_tets_evaluacion	1	1	BAJA
mensaje_error_modificar_informacion_tests_evaluacion	1	1	BAJA
mensaje_ok_modificar_informacion_COBIT	1	1	BAJA
mensaje_error_modificar_informacion_COBIT	1	1	BAJA

Tabla 128. Salidas Externas

7.1.6 Consultas Externas (EQ)

CONSULTA	FTR	DET	COMPLEJIDAD
obtenerProblemas	4	1	BAJA
obtenerTestsEvaluacion	4	1	BAJA
obtenerInformacionCOBIT	5	1	BAJA
obtenerPreferenciasInterfaz	1	1	BAJA
obtenerAyuda	1	1	BAJA
obtenerContraseñaAdministrador	1	1	BAJA
obtenerParametrosBaseDatos	1	1	BAJA
obtenerInformacionProblemas	1	3	BAJA
obtenerInformacionTestsEvaluacion	1	3	BAJA
obtenerInformacionCOBIT	1	5	BAJA

Tabla 129. Consultas Externas

7.1.7 Cálculo de Puntos de Función y SLOC

Descripción	Baja	Media	Alta	Total
Grupo de datos interno (ILF)	28	0	0	28
Grupo de datos externo (EIF)	0	0	0	0
Entradas (EI)	7	3	0	10
Salidas (EO)	15	0	0	15
Consultas (EQ)	10	0	0	10
Total	60	3	0	63

Tabla 130. Cálculos Puntos de Función y SLOC

Para el cálculo de puntos de función se ha utilizado el lenguaje JAVA de COCOMO II 2000, debido al uso de este lenguaje para implementar la aplicación.

SLOC Input Dialog - ITGS

Sizing Method

- ☐ SLOC
- ☒ Function Points
- ☐ Adaptation and Reuse

Breakage
% of code thrown away due to requirements evolution and volatility
REV L

Module Size in Function Points

Language

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Internal Logical Files	<input type="text" value="28"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	196
External Interface Files	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
External Inputs	<input type="text" value="7"/>	<input type="text" value="3"/>	<input type="text" value="0"/>	33
External Outputs	<input type="text" value="15"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	60
External Inquiries	<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	30
Total Unadjusted Function Points				319
Equivalent Total in SLOC				16907

Ilustración 115. Cuadro de diálogo para el cálculo de SLOC

Por lo tanto, es necesario emplear un total de 16907 SLOC.

7.1.8 Factores de Escala

- **PREC** (Desarrollos Previos Familiares): Nominal.
- **FLEX** (Flexibilidad del Desarrollo): Nominal.
- **RESL** (Manejo de Riesgos y Arquitectura): Nominal.
- **TEAM** (Cohesión del Equipo de Desarrollo): Muy Alto.
- **PMAT** (Madurez del Proceso): Nominal.

Se introducen los parámetros en COCOMO II 2000, obteniendo un Staff = 4.6.

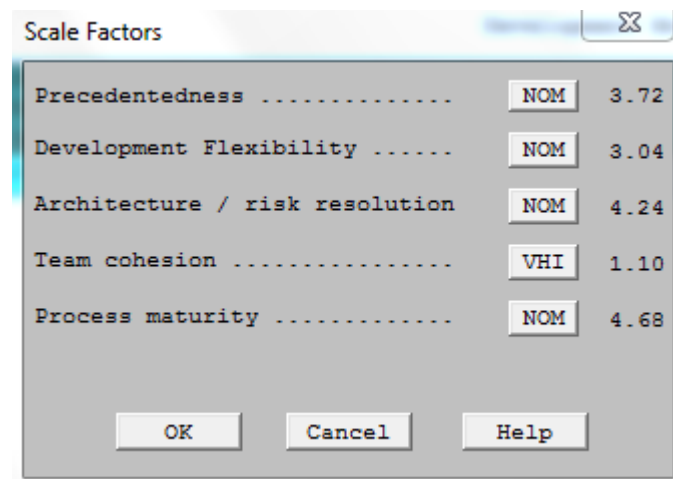


Ilustración 116: Cuadro de diálogo de los Factores de Escala

7.1.9 Drivers de Coste

- ✓ Relacionados con el Producto:
 - **RELY** (Requisitos de Fiabilidad): Nominal.
 - **DATA** (Volumen de la Base de Datos): Nominal.
 - **CPLX** (Complejidad del Producto): Nominal.
 - **RUSE** (Reutilización): Alto.
 - **DOCU** (Documentación Desarrollada): Nominal.
- ✓ Relacionado con la Plataforma de Desarrollo:
 - **TIME** (Exigencias sobre Capacidad de Ejecución): Nominal.
 - **STOR** (Exigencias sobre Almacenamiento de Sistema): Nominal.
 - **PVOL** (Volatilidad de la Plataforma): Nominal.
- ✓ Relacionados con el Personal:
 - **ACAP** (Capacidad de los Analistas): Nominal.

- **PCAP** (Capacidad de los Programadores): Nominal.
- **PLEX** (Experiencia de los Programadores): Nominal.
- **LTEX** (Experiencia con el Lenguaje de Programación): Nominal.
- **APEX** (Experiencia de los Analistas): Nominal.
- **PCON** (Continuidad del Personal): Nominal.

✓ Relacionados con el Proyecto:

- **TOOL** (Uso de Herramientas Software): Nominal.
- **SITE** (Desarrollo en Múltiples Ubicaciones): Nominal.

Se introducen los parámetros en COCOMO II 2000, obteniendo un EAF = 1.07.

The screenshot shows the 'EAF - ITGS' dialog box. It contains several sections with parameters and their ratings. The 'Product' section has ratings for RELY, DATA, DOCU, CPLX, and RUSE. The 'Platform' section has ratings for TIME, STOR, and PVOL. The 'Personnel' section has ratings for ACAP, PCAP, PCON, APEX, LTEX, and PLEX. The 'Project' section has ratings for TOOL and SITE. The 'User' section has ratings for USR1 and USR2. At the bottom, the 'EAF' value is shown as 1.07.

base + Incr % = rating						
Product:	RELY	DATA	DOCU	CPLX	RUSE	
base	NOM	NOM	NOM	NOM	HI	
Incr%	0%	0%	0%	0%	0%	
Platform:	TIME	STOR	PVOL			
base	NOM	NOM	NOM			
Incr%	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	LTEX	PLEX
base	NOM	NOM	NOM	NOM	NOM	NOM
Incr%	0%	0%	0%	0%	0%	0%
Project:	TOOL	SITE				
base	NOM	NOM				
Incr%	0%	0%				
User:	USR1	USR2				
base	NOM	NOM				
Incr%	0%	0%				
EAF is also affected by Schedule						
EAF: 1.07						

Ilustración 117. Ajuste de los Drivers de Coste

El sueldo de los desarrolladores se establece en 1100 euros aproximadamente.

7.1.10 Resultados de la estimación

The screenshot shows the USC-COCOMO II 2000.0 software window. The Project Name is ITGS, and the Development Model is Post Architecture. The main table displays the following data:

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	ITGS	F:16907	1200.00	1.07	JAVA	61.9	66.3	255.1	79528.44	4.7	4.8	0.0

Summary statistics at the bottom:

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	53.0	12.7	318.9	63622.76	3.8	4.2	
Most Likely	66.3	13.7	255.1	79528.44	4.7	4.8	0.0
Pessimistic	82.8	14.7	204.1	99410.55	5.9	5.7	

Total Lines of Code: 16907

Ilustración 118. Resultados de la estimación realizada mediante COCOMO II 2000

COCOMO II 2000 ofrece tres estimaciones:

- **La más optimista** indica que el tiempo necesario sería de 12,7 meses, con un coste de 63.622,76 €.
- **La estimación más probable** indica que el tiempo necesario sería de 13,7 meses, con un coste de 79.528,44 €.
- Y, por último, **la estimación más pesimista** indica que el tiempo necesario sería de 14,7 meses, con un coste de 99.410,55 €.

Al ser este un proyecto realizado por dos personas, el coste estimado mediante COCOMO II 2000 sería la mitad del valor obtenido para cada persona.

7.2 Costes reales

7.2.1 Plan de trabajo: WBS

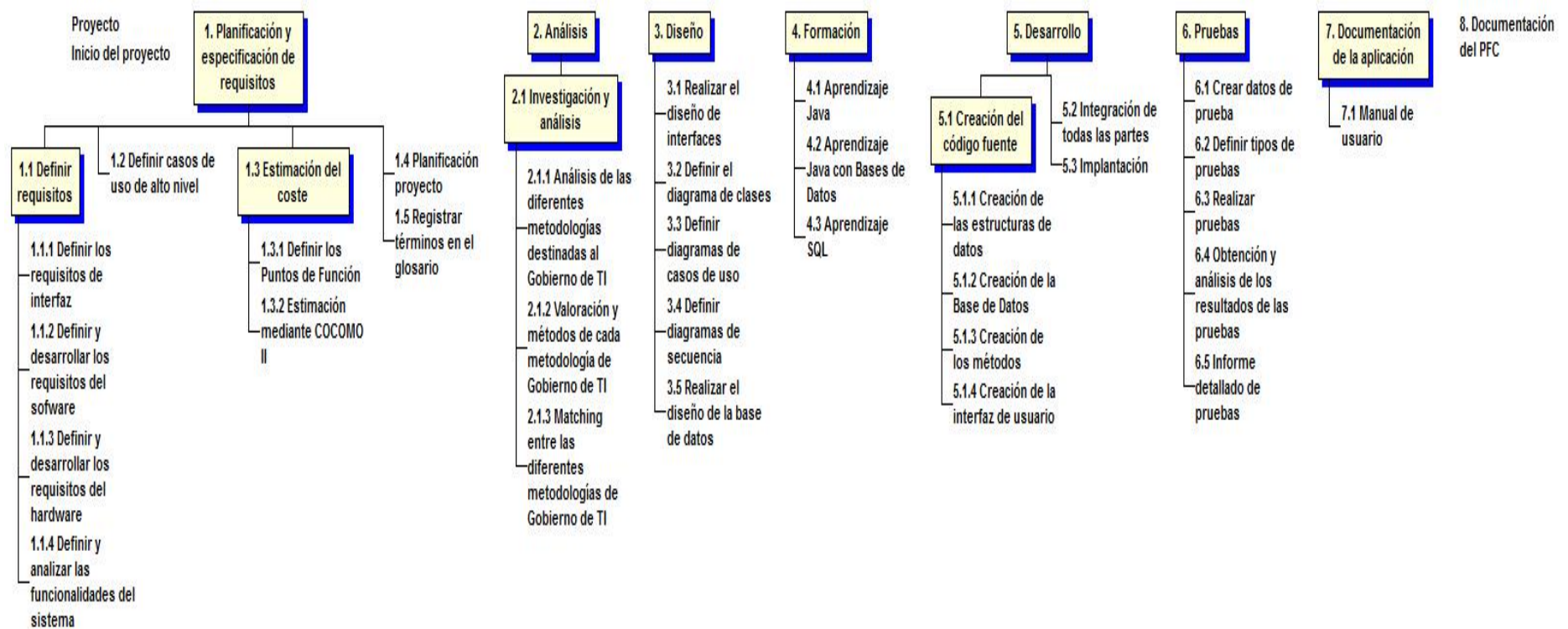


Ilustración 119. Plan de trabajo: WBS

7.2.2 Cronograma de planificación real



Ilustración 120. Diagrama de Gantt – Parte (1/4)

Análisis, Diseño e Implementación de una herramienta de resolución de problemas en objetivos de TI

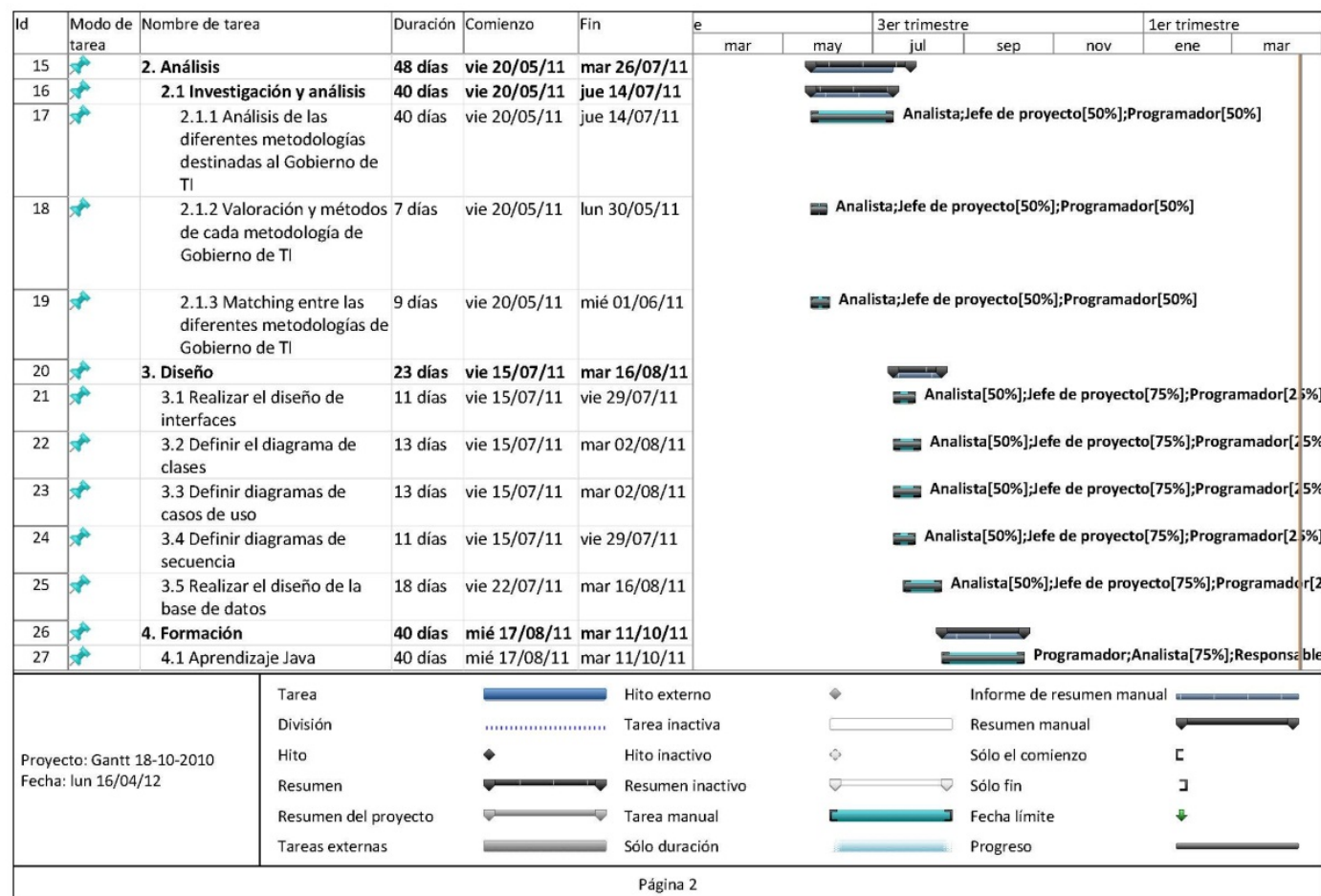


Ilustración 121. Diagrama de Gantt – Parte (2/4)

Análisis, Diseño e Implementación de una herramienta de resolución de problemas en objetivos de TI

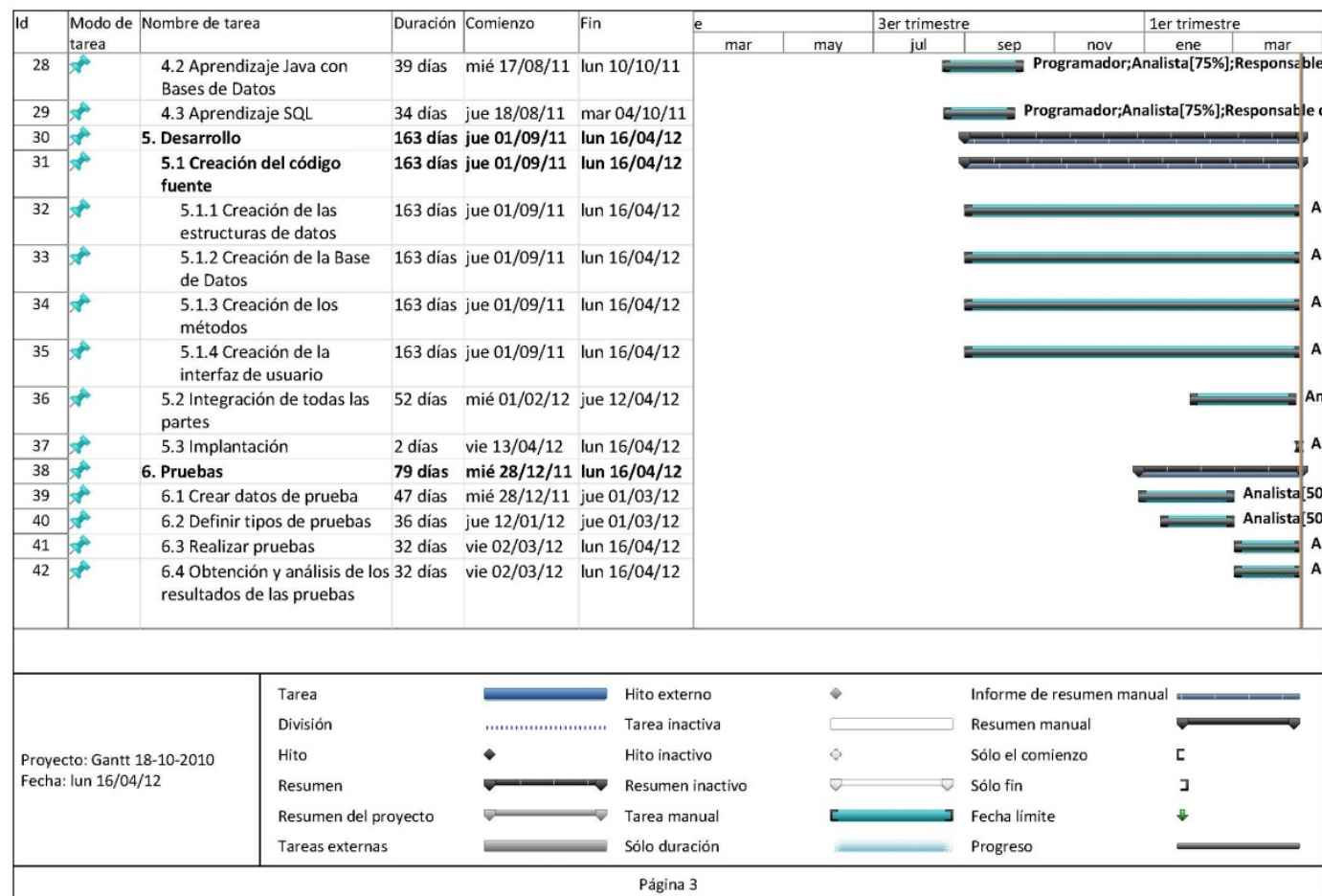


Ilustración 122. Diagrama de Gantt – Parte (3/4)

Análisis, Diseño e Implementación de una herramienta de resolución de problemas en objetivos de TI



Ilustración 123. Diagrama de Gantt – Parte (4/4)

7.2.3 Tareas y recursos humanos

Tareas y recursos humanos el lun 16/04/12
Gantt 18-10-2010

Id	Indicador	Nombre del recurso	Trabajo				
1			0 horas				
2		Analista	9.183,6 horas				
Identificador	Nombre de tarea	Unidades	Trabajo	Retraso	Comienzo	Fin	
3	1. Planificación y especificación de requisitos	100%	456 horas	0 días	mié 02/03/11	jue 19/05/11	
15	2. Análisis	100%	384 horas	0 días	vie 20/05/11	mar 26/07/11	
4	1.1 Definir requisitos	100%	264 horas	0 días	mié 02/03/11	vie 15/04/11	
5	1.1.1 Definir los requisitos de interfaz	100%	120 horas	0 días	mié 02/03/11	mar 22/03/11	
6	1.1.2 Definir y desarrollar los requisitos del software	100%	264 horas	0 días	mié 02/03/11	vie 15/04/11	
7	1.1.3 Definir y desarrollar los requisitos del hardware	100%	80 horas	0 días	mié 02/03/11	mar 15/03/11	
8	1.1.4 Definir y analizar las funcionalidades del sistema	100%	264 horas	0 días	mié 02/03/11	vie 15/04/11	
9	1.2 Definir casos de uso de alto nivel	100%	192 horas	0 días	mié 02/03/11	lun 04/04/11	
10	1.3 Estimación del coste	100%	112 horas	0 días	vie 18/03/11	mié 06/04/11	
11	1.3.1 Definir los Puntos de Función	100%	24 horas	0 días	vie 18/03/11	mar 22/03/11	
12	1.3.2 Estimación mediante COCOMO II	100%	88 horas	0 días	vie 18/03/11	vie 01/04/11	
13	1.4 Planificación proyecto	100%	56 horas	0 días	vie 18/03/11	lun 28/03/11	
14	1.5 Registrar términos en el glosario	100%	360 horas	0 días	vie 18/03/11	jue 19/05/11	
16	2.1 Investigación y análisis	100%	320 horas	0 días	vie 20/05/11	jue 14/07/11	
17	2.1.1 Análisis de las diferentes metodologías destinadas al Gobierno de TI	100%	320 horas	0 días	vie 20/05/11	jue 14/07/11	
18	2.1.2 Valoración y métodos de cada metodología de Gobierno de TI	100%	56 horas	0 días	vie 20/05/11	lun 30/05/11	
19	2.1.3 Matching entre las diferentes metodologías de Gobierno de TI	100%	72 horas	0 días	vie 20/05/11	mié 01/06/11	
20	3. Diseño	50%	92 horas	0 días	vie 15/07/11	mar 16/08/11	
21	3.1 Realizar el diseño de interfaces	50%	44 horas	0 días	vie 15/07/11	vie 29/07/11	
22	3.2 Definir el diagrama de clases	50%	52 horas	0 días	vie 15/07/11	mar 02/08/11	
23	3.3 Definir diagramas de casos de uso	50%	52 horas	0 días	vie 15/07/11	mar 02/08/11	
24	3.4 Definir diagramas de secuencia	50%	44 horas	0 días	vie 15/07/11	vie 29/07/11	
25	3.5 Realizar el diseño de la base de datos	50%	72 horas	0 días	vie 22/07/11	mar 16/08/11	
26	4. Formación	75%	240 horas	0 días	mié 17/08/11	mar 11/10/11	
27	4.1 Aprendizaje Java	75%	240 horas	0 días	mié 17/08/11	mar 11/10/11	
28	4.2 Aprendizaje Java con Bases de Datos	75%	234 horas	0 días	mié 17/08/11	lun 10/10/11	
29	4.3 Aprendizaje SQL	75%	204 horas	0 días	jue 18/08/11	mar 04/10/11	
30	5. Desarrollo	10%	130,4 horas	0 días	jue 01/09/11	lun 16/04/12	
31	5.1 Creación del código fuente	10%	130,4 horas	0 días	jue 01/09/11	lun 16/04/12	
32	5.1.1 Creación de las estructuras de datos	10%	130,4 horas	0 días	jue 01/09/11	lun 16/04/12	
33	5.1.2 Creación de la Base de Datos	10%	130,4 horas	0 días	jue 01/09/11	lun 16/04/12	
34	5.1.3 Creación de los métodos	10%	130,4 horas	0 días	jue 01/09/11	lun 16/04/12	
35	5.1.4 Creación de la interfaz de usuario	10%	130,4 horas	0 días	jue 01/09/11	lun 16/04/12	
36	5.2 Integración de todas las partes	10%	41,6 horas	0 días	mié 01/02/12	jue 12/04/12	
37	5.3 Implantación	10%	1,6 horas	0 días	vie 13/04/12	lun 16/04/12	
38	6. Pruebas	50%	316 horas	0 días	mié 28/12/11	lun 16/04/12	
39	6.1 Crear datos de prueba	50%	188 horas	0 días	mié 28/12/11	jue 01/03/12	
40	6.2 Definir tipos de pruebas	50%	144 horas	0 días	jue 12/01/12	jue 01/03/12	
41	6.3 Realizar pruebas	50%	128 horas	0 días	vie 02/03/12	lun 16/04/12	
42	6.4 Obtención y análisis de los resultados de las pruebas	50%	128 horas	0 días	vie 02/03/12	lun 16/04/12	
43	6.5 Informe detallado de pruebas	50%	4 horas	0 días	lun 16/04/12	lun 16/04/12	
44	7. Documentación de la aplicación	50%	196 horas	0 días	mié 08/02/12	lun 16/04/12	
45	7.1 Manual de usuario	50%	196 horas	0 días	mié 08/02/12	lun 16/04/12	
46	8. Documentación del PFC	100%	2.352 horas	0 días	mié 02/03/11	lun 16/04/12	

Ilustración 124. Tareas y recursos humanos – Parte (1/4)

Análisis, Diseño e Implementación de una herramienta de resolución de problemas en objetivos de TI

Tareas y recursos humanos el lun 16/04/12
Gantt 18-10-2010

Id	Indicador	Nombre del recurso	Trabajo			
3	Programador	14.736,8 horas				
Identificador	Nombre de tarea	Unidades	Trabajo	Retraso	Comienzo	Fin
15	2. Análisis	50%	192 horas	0 días	vie 20/05/11	mar 26/07/11
3	1. Planificación y especificación de requisitos	25%	114 horas	0 días	mié 02/03/11	jue 19/05/11
4	1.1 Definir requisitos	25%	66 horas	0 días	mié 02/03/11	vie 15/04/11
5	1.1.1 Definir los requisitos de interfaz	25%	30 horas	0 días	mié 02/03/11	mar 22/03/11
6	1.1.2 Definir y desarrollar los requisitos del software	25%	66 horas	0 días	mié 02/03/11	vie 15/04/11
7	1.1.3 Definir y desarrollar los requisitos del hardware	25%	20 horas	0 días	mié 02/03/11	mar 15/03/11
8	1.1.4 Definir y analizar las funcionalidades del sistema	25%	66 horas	0 días	mié 02/03/11	vie 15/04/11
9	1.2 Definir casos de uso de alto nivel	25%	48 horas	0 días	mié 02/03/11	lun 04/04/11
10	1.3 Estimación del coste	25%	28 horas	0 días	vie 18/03/11	mié 06/04/11
11	1.3.1 Definir los Puntos de Función	25%	6 horas	0 días	vie 18/03/11	mar 22/03/11
12	1.3.2 Estimación mediante COCOMO II	25%	22 horas	0 días	vie 18/03/11	vie 01/04/11
13	1.4 Planificación proyecto	25%	14 horas	0 días	vie 18/03/11	lun 28/03/11
14	1.5 Registrar términos en el glosario	25%	90 horas	0 días	vie 18/03/11	jue 19/05/11
16	2.1 Investigación y análisis	50%	160 horas	0 días	vie 20/05/11	jue 14/07/11
17	2.1.1 Análisis de las diferentes metodologías destinadas al Gobierno de TI	50%	160 horas	0 días	vie 20/05/11	jue 14/07/11
18	2.1.2 Valoración y métodos de cada metodología de Gobierno de TI	50%	28 horas	0 días	vie 20/05/11	lun 30/05/11
19	2.1.3 Matching entre las diferentes metodologías de Gobierno de TI	50%	36 horas	0 días	vie 20/05/11	mié 01/06/11
20	3. Diseño	25%	46 horas	0 días	vie 15/07/11	mar 16/08/11
21	3.1 Realizar el diseño de interfaces	25%	22 horas	0 días	vie 15/07/11	vie 29/07/11
22	3.2 Definir el diagrama de clases	25%	26 horas	0 días	vie 15/07/11	mar 02/08/11
23	3.3 Definir diagramas de casos de uso	25%	26 horas	0 días	vie 15/07/11	mar 02/08/11
24	3.4 Definir diagramas de secuencia	25%	22 horas	0 días	vie 15/07/11	vie 29/07/11
25	3.5 Realizar el diseño de la base de datos	25%	36 horas	0 días	vie 22/07/11	mar 16/08/11
26	4. Formación	100%	320 horas	0 días	mié 17/08/11	mar 11/10/11
27	4.1 Aprendizaje Java	100%	320 horas	0 días	mié 17/08/11	mar 11/10/11
28	4.2 Aprendizaje Java con Bases de Datos	100%	312 horas	0 días	mié 17/08/11	lun 10/10/11
29	4.3 Aprendizaje SQL	100%	272 horas	0 días	jue 18/08/11	mar 04/10/11
30	5. Desarrollo	100%	1.304 horas	0 días	jue 01/09/11	lun 16/04/12
31	5.1 Creación del código fuente	100%	1.304 horas	0 días	jue 01/09/11	lun 16/04/12
32	5.1.1 Creación de las estructuras de datos	100%	1.304 horas	0 días	jue 01/09/11	lun 16/04/12
33	5.1.2 Creación de la Base de Datos	100%	1.304 horas	0 días	jue 01/09/11	lun 16/04/12
34	5.1.3 Creación de los métodos	100%	1.304 horas	0 días	jue 01/09/11	lun 16/04/12
35	5.1.4 Creación de la interfaz de usuario	100%	1.304 horas	0 días	jue 01/09/11	lun 16/04/12
36	5.2 Integración de todas las partes	100%	416 horas	0 días	mié 01/02/12	jue 12/04/12
37	5.3 Implantación	100%	16 horas	0 días	vie 13/04/12	lun 16/04/12
38	6. Pruebas	100%	632 horas	0 días	mié 28/12/11	lun 16/04/12
39	6.1 Crear datos de prueba	100%	376 horas	0 días	mié 28/12/11	jue 01/03/12
40	6.2 Definir tipos de pruebas	100%	288 horas	0 días	jue 12/01/12	jue 01/03/12
41	6.3 Realizar pruebas	100%	256 horas	0 días	vie 02/03/12	lun 16/04/12
42	6.4 Obtención y análisis de los resultados de las pruebas	100%	256 horas	0 días	vie 02/03/12	lun 16/04/12
43	6.5 Informe detallado de pruebas	100%	8 horas	0 días	lun 16/04/12	lun 16/04/12
44	7. Documentación de la aplicación	60%	235,2 horas	0 días	mié 08/02/12	lun 16/04/12
45	7.1 Manual de usuario	60%	235,2 horas	0 días	mié 08/02/12	lun 16/04/12
46	8. Documentación del PFC	70%	1.646,4 horas	0 días	mié 02/03/11	lun 16/04/12

Ilustración 125. Tareas y recursos humanos – Parte (2/4)

Análisis, Diseño e Implementación de una herramienta de resolución de problemas en objetivos de TI

Tareas y recursos humanos el lun 16/04/12
Gantt 18-10-2010

Id	Indicadore	Nombre del recurso	Trabajo				
4	Responsable de calidad		14.227,6 horas				
	Identificador	Nombre de tarea	Unidades	Trabajo	Retraso	Comienzo	Fin
	3	1. Planificación y especificación de requisitos	75%	342 horas	0 días	mié 02/03/11	jue 19/05/11
	4	1.1 Definir requisitos	75%	198 horas	0 días	mié 02/03/11	vie 15/04/11
	5	1.1.1 Definir los requisitos de interfaz	75%	90 horas	0 días	mié 02/03/11	mar 22/03/11
	6	1.1.2 Definir y desarrollar los requisitos del software	75%	198 horas	0 días	mié 02/03/11	vie 15/04/11
	7	1.1.3 Definir y desarrollar los requisitos del hardware	75%	60 horas	0 días	mié 02/03/11	mar 15/03/11
	8	1.1.4 Definir y analizar las funcionalidades del sistema	75%	198 horas	0 días	mié 02/03/11	vie 15/04/11
	9	1.2 Definir casos de uso de alto nivel	75%	144 horas	0 días	mié 02/03/11	lun 04/04/11
	10	1.3 Estimación del coste	75%	84 horas	0 días	vie 18/03/11	mié 06/04/11
	11	1.3.1 Definir los Puntos de Función	75%	18 horas	0 días	vie 18/03/11	mar 22/03/11
	12	1.3.2 Estimación mediante COCOMO II	75%	66 horas	0 días	vie 18/03/11	vie 01/04/11
	13	1.4 Planificación proyecto	75%	42 horas	0 días	vie 18/03/11	lun 28/03/11
	14	1.5 Registrar términos en el glosario	75%	270 horas	0 días	vie 18/03/11	jue 19/05/11
	20	3. Diseño	100%	184 horas	0 días	vie 15/07/11	mar 16/08/11
	21	3.1 Realizar el diseño de interfaces	100%	88 horas	0 días	vie 15/07/11	vie 29/07/11
	22	3.2 Definir el diagrama de clases	100%	104 horas	0 días	vie 15/07/11	mar 02/08/11
	23	3.3 Definir diagramas de casos de uso	100%	104 horas	0 días	vie 15/07/11	mar 02/08/11
	24	3.4 Definir diagramas de secuencia	100%	88 horas	0 días	vie 15/07/11	vie 29/07/11
	25	3.5 Realizar el diseño de la base de datos	100%	144 horas	0 días	vie 22/07/11	mar 16/08/11
	26	4. Formación	50%	160 horas	0 días	mié 17/08/11	mar 11/10/11
	27	4.1 Aprendizaje Java	50%	160 horas	0 días	mié 17/08/11	mar 11/10/11
	28	4.2 Aprendizaje Java con Bases de Datos	50%	156 horas	0 días	mié 17/08/11	lun 10/10/11
	29	4.3 Aprendizaje SQL	50%	136 horas	0 días	jue 18/08/11	mar 04/10/11
	30	5. Desarrollo	80%	1.043,2 horas	0 días	jue 01/09/11	lun 16/04/12
	31	5.1 Creación del código fuente	80%	1.043,2 horas	0 días	jue 01/09/11	lun 16/04/12
	32	5.1.1 Creación de las estructuras de datos	80%	1.043,2 horas	0 días	jue 01/09/11	lun 16/04/12
	33	5.1.2 Creación de la Base de Datos	80%	1.043,2 horas	0 días	jue 01/09/11	lun 16/04/12
	34	5.1.3 Creación de los métodos	80%	1.043,2 horas	0 días	jue 01/09/11	lun 16/04/12
	35	5.1.4 Creación de la interfaz de usuario	80%	1.043,2 horas	0 días	jue 01/09/11	lun 16/04/12
	36	5.2 Integración de todas las partes	80%	332,8 horas	0 días	mié 01/02/12	jue 12/04/12
	37	5.3 Implantación	80%	12,8 horas	0 días	vie 13/04/12	lun 16/04/12
	38	6. Pruebas	80%	505,6 horas	0 días	mié 28/12/11	lun 16/04/12
	39	6.1 Crear datos de prueba	80%	300,8 horas	0 días	mié 28/12/11	jue 01/03/12
	40	6.2 Definir tipos de pruebas	80%	230,4 horas	0 días	jue 12/01/12	jue 01/03/12
	41	6.3 Realizar pruebas	80%	204,8 horas	0 días	vie 02/03/12	lun 16/04/12
	42	6.4 Obtención y análisis de los resultados de las pruebas	80%	204,8 horas	0 días	vie 02/03/12	lun 16/04/12
	43	6.5 Informe detallado de pruebas	80%	6,4 horas	0 días	lun 16/04/12	lun 16/04/12
	44	7. Documentación de la aplicación	100%	392 horas	0 días	mié 08/02/12	lun 16/04/12
	45	7.1 Manual de usuario	100%	392 horas	0 días	mié 08/02/12	lun 16/04/12
	46	8. Documentación del PFC	100%	2.352 horas	0 días	mié 02/03/11	lun 16/04/12
5	Jefe de proyecto		12.300,8 horas				
	Identificador	Nombre de tarea	Unidades	Trabajo	Retraso	Comienzo	Fin
	3	1. Planificación y especificación de requisitos	75%	342 horas	0 días	mié 02/03/11	jue 19/05/11
	15	2. Análisis	50%	192 horas	0 días	vie 20/05/11	mar 26/07/11
	4	1.1 Definir requisitos	75%	198 horas	0 días	mié 02/03/11	vie 15/04/11
	5	1.1.1 Definir los requisitos de interfaz	75%	90 horas	0 días	mié 02/03/11	mar 22/03/11

Ilustración 126. Tareas y recursos humanos – Parte (3/4)

Análisis, Diseño e Implementación de una herramienta de resolución de problemas en objetivos de TI

Tareas y recursos humanos el lun 16/04/12
Gantt 18-10-2010

Id	Indicadore	Nombre del recurso	Trabajo				
"Jefe de proyecto" (continuación)							
Identificador	Nombre de tarea	Unidades	Trabajo	Retraso	Comienzo	Fin	
6	1.1.2 Definir y desarrollar los requisitos del software	75%	198 horas	0 días	mié 02/03/11	vie 15/04/11	
7	1.1.3 Definir y desarrollar los requisitos del hardware	75%	60 horas	0 días	mié 02/03/11	mar 15/03/11	
8	1.1.4 Definir y analizar las funcionalidades del sistema	75%	198 horas	0 días	mié 02/03/11	vie 15/04/11	
9	1.2 Definir casos de uso de alto nivel	75%	144 horas	0 días	mié 02/03/11	lun 04/04/11	
10	1.3 Estimación del coste	75%	84 horas	0 días	vie 18/03/11	mié 06/04/11	
11	1.3.1 Definir los Puntos de Función	75%	18 horas	0 días	vie 18/03/11	mar 22/03/11	
12	1.3.2 Estimación mediante COCOMO II	75%	66 horas	0 días	vie 18/03/11	vie 01/04/11	
13	1.4 Planificación proyecto	75%	42 horas	0 días	vie 18/03/11	lun 28/03/11	
14	1.5 Registrar términos en el glosario	75%	270 horas	0 días	vie 18/03/11	jue 19/05/11	
16	2.1 Investigación y análisis	50%	160 horas	0 días	vie 20/05/11	jue 14/07/11	
17	2.1.1 Análisis de las diferentes metodologías destinadas al Gobierno de TI	50%	160 horas	0 días	vie 20/05/11	jue 14/07/11	
18	2.1.2 Valoración y métodos de cada metodología de Gobierno de TI	50%	28 horas	0 días	vie 20/05/11	lun 30/05/11	
19	2.1.3 Matching entre las diferentes metodologías de Gobierno de TI	50%	36 horas	0 días	vie 20/05/11	mié 01/06/11	
20	3. Diseño	75%	138 horas	0 días	vie 15/07/11	mar 16/08/11	
21	3.1 Realizar el diseño de interfaces	75%	66 horas	0 días	vie 15/07/11	vie 29/07/11	
22	3.2 Definir el diagrama de clases	75%	78 horas	0 días	vie 15/07/11	mar 02/08/11	
23	3.3 Definir diagramas de casos de uso	75%	78 horas	0 días	vie 15/07/11	mar 02/08/11	
24	3.4 Definir diagramas de secuencia	75%	66 horas	0 días	vie 15/07/11	vie 29/07/11	
25	3.5 Realizar el diseño de la base de datos	75%	108 horas	0 días	vie 22/07/11	mar 16/08/11	
30	5. Desarrollo	70%	912,8 horas	0 días	jue 01/09/11	lun 16/04/12	
31	5.1 Creación del código fuente	70%	912,8 horas	0 días	jue 01/09/11	lun 16/04/12	
32	5.1.1 Creación de las estructuras de datos	70%	912,8 horas	0 días	jue 01/09/11	lun 16/04/12	
33	5.1.2 Creación de la Base de Datos	70%	912,8 horas	0 días	jue 01/09/11	lun 16/04/12	
34	5.1.3 Creación de los métodos	70%	912,8 horas	0 días	jue 01/09/11	lun 16/04/12	
35	5.1.4 Creación de la interfaz de usuario	70%	912,8 horas	0 días	jue 01/09/11	lun 16/04/12	
36	5.2 Integración de todas las partes	70%	291,2 horas	0 días	mié 01/02/12	jue 12/04/12	
37	5.3 Implantación	70%	11,2 horas	0 días	vie 13/04/12	lun 16/04/12	
38	6. Pruebas	70%	442,4 horas	0 días	mié 28/12/11	lun 16/04/12	
39	6.1 Crear datos de prueba	70%	263,2 horas	0 días	mié 28/12/11	jue 01/03/12	
40	6.2 Definir tipos de pruebas	70%	201,6 horas	0 días	jue 12/01/12	jue 01/03/12	
41	6.3 Realizar pruebas	70%	179,2 horas	0 días	vie 02/03/12	lun 16/04/12	
42	6.4 Obtención y análisis de los resultados de las pruebas	70%	179,2 horas	0 días	vie 02/03/12	lun 16/04/12	
43	6.5 Informe detallado de pruebas	70%	5,6 horas	0 días	lun 16/04/12	lun 16/04/12	
44	7. Documentación de la aplicación	70%	274,4 horas	0 días	mié 08/02/12	lun 16/04/12	
45	7.1 Manual de usuario	70%	274,4 horas	0 días	mié 08/02/12	lun 16/04/12	
46	8. Documentación del PFC	80%	1.881,6 horas	0 días	mié 02/03/11	lun 16/04/12	

Ilustración 127. Tareas y recursos humanos – Parte (4/4)

7.2.4 Informe presupuestario

Informe presupuestario el lun 16/04/12
Gantt 18-10-2010

Id	Nombre de tarea	Costo fijo	Acumulación de costos fijos	Costo total	Previsto	Variación	Real
46	8. Documentación del PFC	0,00 €	Prorratio	47.280,80 €	0,00 €	47.280,80 €	47.280,80 €
32	5.1.1 Creación de las estructuras	0,00 €	Prorratio	19.828,80 €	0,00 €	19.828,80 €	19.828,80 €
33	5.1.2 Creación de la Base de Datos	0,00 €	Prorratio	19.828,80 €	0,00 €	19.828,80 €	19.828,80 €
34	5.1.3 Creación de los métodos	0,00 €	Prorratio	19.828,80 €	0,00 €	19.828,80 €	19.828,80 €
35	5.1.4 Creación de la interfaz de usuario	0,00 €	Prorratio	19.828,80 €	0,00 €	19.828,80 €	19.828,80 €
39	6.1 Crear datos de prueba	0,00 €	Prorratio	6.475,20 €	0,00 €	6.475,20 €	6.475,20 €
45	7.1 Manual de usuario	0,00 €	Prorratio	6.433,60 €	0,00 €	6.433,60 €	6.433,60 €
36	5.2 Integración de todas las partes	0,00 €	Prorratio	6.331,20 €	0,00 €	6.331,20 €	6.331,20 €
14	1.5 Registrar términos en el glosario	0,00 €	Prorratio	5.762,00 €	0,00 €	5.762,00 €	5.762,00 €
40	6.2 Definir tipos de pruebas	0,00 €	Prorratio	4.961,60 €	0,00 €	4.961,60 €	4.961,60 €
41	6.3 Realizar pruebas	0,00 €	Prorratio	4.411,20 €	0,00 €	4.411,20 €	4.411,20 €
42	6.4 Obtención y análisis de los requisitos	0,00 €	Prorratio	4.411,20 €	0,00 €	4.411,20 €	4.411,20 €
6	1.1.2 Definir y desarrollar los requisitos	0,00 €	Prorratio	4.226,00 €	0,00 €	4.226,00 €	4.226,00 €
8	1.1.4 Definir y analizar las funciones	0,00 €	Prorratio	4.226,00 €	0,00 €	4.226,00 €	4.226,00 €
27	4.1 Aprendizaje Java	0,00 €	Prorratio	3.768,00 €	0,00 €	3.768,00 €	3.768,00 €
28	4.2 Aprendizaje Java con Bases de Datos	0,00 €	Prorratio	3.674,00 €	0,00 €	3.674,00 €	3.674,00 €
17	2.1.1 Análisis de las diferentes metodologías	0,00 €	Prorratio	3.524,00 €	0,00 €	3.524,00 €	3.524,00 €
29	4.3 Aprendizaje SQL	0,00 €	Prorratio	3.204,00 €	0,00 €	3.204,00 €	3.204,00 €
9	1.2 Definir casos de uso de alto nivel	0,00 €	Prorratio	3.074,00 €	0,00 €	3.074,00 €	3.074,00 €
25	3.5 Realizar el diseño de la base de datos	0,00 €	Prorratio	2.162,00 €	0,00 €	2.162,00 €	2.162,00 €
5	1.1.1 Definir los requisitos de interfaz	0,00 €	Prorratio	1.922,00 €	0,00 €	1.922,00 €	1.922,00 €
22	3.2 Definir el diagrama de clases	0,00 €	Prorratio	1.562,00 €	0,00 €	1.562,00 €	1.562,00 €
23	3.3 Definir diagramas de casos de uso	0,00 €	Prorratio	1.562,00 €	0,00 €	1.562,00 €	1.562,00 €
12	1.3.2 Estimación mediante COCOMO	0,00 €	Prorratio	1.410,00 €	0,00 €	1.410,00 €	1.410,00 €
21	3.1 Realizar el diseño de interfaz	0,00 €	Prorratio	1.322,00 €	0,00 €	1.322,00 €	1.322,00 €
24	3.4 Definir diagramas de secuencia	0,00 €	Prorratio	1.322,00 €	0,00 €	1.322,00 €	1.322,00 €
7	1.1.3 Definir y desarrollar los requisitos	0,00 €	Prorratio	1.282,00 €	0,00 €	1.282,00 €	1.282,00 €
13	1.4 Planificación proyecto	0,00 €	Prorratio	898,00 €	0,00 €	898,00 €	898,00 €
19	2.1.3 Matching entre las diferentes metodologías	0,00 €	Prorratio	796,00 €	0,00 €	796,00 €	796,00 €
18	2.1.2 Valoración y métodos de comparación	0,00 €	Prorratio	620,00 €	0,00 €	620,00 €	620,00 €
11	1.3.1 Definir los Puntos de Función	0,00 €	Prorratio	386,00 €	0,00 €	386,00 €	386,00 €
37	5.3 Implantación	0,00 €	Prorratio	251,20 €	0,00 €	251,20 €	251,20 €
43	6.5 Informe detallado de pruebas	0,00 €	Prorratio	145,60 €	0,00 €	145,60 €	145,60 €
1	Proyecto	0,00 €	Prorratio	0,00 €	0,00 €	0,00 €	0,00 €

Ilustración 128. Informe presupuestario – Parte (1/2)

Informe presupuestario el lun 16/04/12
Gantt 18-10-2010

Id	Nombre de tarea	Costo fijo	Acumulación de costos fijos	Costo total	Previsto	Variación	Real
2	Inicio del proyecto	0,00 €	Prorrateo	0,00 €	0,00 €	0,00 €	0,00 €
		0,00 €		206.718,80 €	0,00 €	206.718,80 €	206.718,80 €

Ilustración 129. Informe presupuestario – Parte (2/2)

Al ser este un proyecto realizado por dos personas, el coste y el trabajo real serían la mitad de los valores obtenidos para cada persona.

8. Líneas futuras

Durante la realización del proyecto surgieron algunas posibles ideas que se podrían llevar a cabo para seguir incrementando la capacidad del aplicativo pero que por falta de tiempo o medios no se llevaron finalmente a cabo, y es por ello que se recogen en este apartado de posibles líneas futuras.

Algunas de estas ideas se han añadido al programa a modo de demostración pero sin llegar a estar finalizadas completamente. Tales como el idioma y la impresión desde la propia aplicación.

El programa está realizado en inglés por su internacionalización y el gran volumen existente de documentación sobre gobierno TI en este idioma. Sin embargo sería posible una traducción al español u otros idiomas. A modo de prueba desde el menú del programa podemos cambiar el idioma del inglés al español y viceversa pero solo en unas pocas ventanas del programa, simplemente como demostración. La principal razón de no completar el cambio del idioma fue porque supondría añadir una cantidad relativamente grande de código para el idioma y sobretodo porque supondría duplicar la base de datos en dos, una en inglés y otra en español, lo cual haría necesario añadir un atributo de idioma a cada tabla de la base de datos y cambiar las consultas, borrados e inserciones que se realizan en la aplicación añadiendo una cláusula más para elegir el idioma. Además se habrían tenido que realizar algunos cambios y controles propios del idioma con los caracteres.

El manual de usuario está disponible en español pero no en inglés. Si hacemos *click* en el idioma español, el programa cambiará a español y si pulsamos en la ayuda y a continuación en el manual de usuario nos aparecerá un fichero en formato PDF del manual de usuario en español. Si por el contrario elegimos utilizar el programa en inglés y pulsamos sobre la guía de usuario se nos abrirá un archivo PDF indicándonos que está en construcción y que tenemos a nuestra disposición una versión en español. Únicamente con cambiar el fichero existente llamado 'UserGuideITGS.pdf' por uno que esté completo en inglés el programa funcionaría correctamente con su correspondiente manual de usuario en inglés.

La impresión en papel que se puede realizar desde el propio programa solo permite imprimir una página y no varias. La impresión presentó algunos problemas durante la implementación en código y se decidió dejar la demostración de una hoja, ya que por otro lado el programa permite la exportación a ficheros de texto desde los cuales se puede imprimir sin problema.

En cuanto a la seguridad del programa, sería conveniente realizar un cifrado más seguro de la contraseña del administrador. Hubo algunos problemas con los caracteres en el tratamiento de cifrado por lo que ahora mismo el cifrado es algo simple. También sería una opción realizar un tratamiento especial para los archivos que se usan para configuraciones del programa tales como cifrarlos, ocultarlos o

prohibir su borrado. Hacer un registro en un fichero de posibles errores que hayan surgido durante la ejecución del programa sería beneficioso para la mejora de la herramienta o ayuda al usuario que estuviera haciendo un mal uso cambiando algún aspecto problemático y sería otra característica que podría añadirse a la aplicación, incluso si estuviera funcionando en modo de applet a través de una web podrían enviarse estos informes de errores vía online.

Otras ideas que surgieron durante la realización del proyecto fueron el de la integración del aplicativo en una página web a través de un Applet de Java y PHP. O del desarrollo, ya que está realizado en Java, en un sistema operativo Android (la programación en este sistema operativo móvil se realiza a través de un lenguaje similar a Java aunque con algunas importantes variantes) para consultas desde el móvil.

También sería una buena opción hacer un instalador para la aplicación junto de la base de datos o un migrado de la base de datos para su correcto funcionamiento en Linux y otros sistemas operativos, y así facilitar la tarea de la instalación para hacerlo de una manera más automática y compatible en sistemas operativos heterogéneos.

En el menú de preferencias del programa se permiten cambiar ciertas opciones a gusto del usuario. Pero se podrían incrementar las opciones como cambio de tamaño de botones y sus letras para facilitar el uso a personas con problemas de accesibilidad de visión o cambiando el color a los botones, fondos y texto... e incluso añadiendo sonido que lea la información con el paso del ratón por encima de los distintos elementos como botones, etiquetas... añadir una lupa de aumento o *zoom* para facilitar la lectura en ciertos sitios, un mayor número de tooltips, etc. Se podrían añadir gráficos pequeños en la barra de menú que fueran fácilmente identificables como los típicos botones en las barras de los programas para exportar archivos, impresión...

Se podría realizar también un aumento en el número de caracteres permitidos en el tratamiento de caracteres del usuario para evitar el problema de la inyección SQL, aunque con los caracteres actuales se puede describir cualquier tipo de información con la que se trabaja actualmente en la aplicación.

También cabe decir que llevar un mantenimiento continuo de datos en la aplicación, una actualización, es decir dar de alta nuevos problemas y soluciones hará que cuanto mayor sea la base de datos mayor probabilidad de encontrar solución a mayor número de problemas, y por tanto más fiable y útil se mostrará el programa. En particular la introducción de nuevos objetivos TI (no disponibles explícitamente en COBIT y por tanto no introducidos en la base de datos) y sus relaciones con los objetivos de negocio (ya existentes en la base de datos) permitirían la obtención de una solución más completa mostrando ambos objetivos.

Muchas de estas ideas surgieron por la posibilidad de llevar este proyecto hacia un paso más y hacer de él un uso real en el mundo TI. Podrían realizarse por tanto muchas mejoras más para hacer de él un programa más competitivo y comercial pero

su desarrollo ya se vería fuera de la misión de un proyecto fin de carrera. Es por ello que aquí se dan únicamente algunas pinceladas para su posible mejora en un futuro.

9. Conclusiones

La realización del proyecto ha supuesto el cumplimiento de todos los objetivos planteados en el apartado de “Objetivos del proyecto” por un lado el uso de los conocimientos adquiridos durante la carrera y la adquisición de nuevos conocimientos muy demandados en el mundo laboral, como son los relacionados con el lenguaje de programación Java y su uso conjunto con bases de datos y las diferentes metodologías dentro del ámbito del Gobierno de las Tecnologías de la Información, en especial el marco de trabajo COBIT.

El aprendizaje de Java se ha hecho desde cero y de forma autodidacta ya que no se tenían conocimientos previos sobre este lenguaje, así como también algunos aspectos en la base de datos, como el tratamiento de gran cantidad de caracteres o la conexión entre la base de datos y el lenguaje Java. Sobre el aprendizaje acerca de las metodologías del Gobierno de las TI tampoco se tenían grandes conocimientos y de su estudio se ha obtenido una mayor visión del mundo de las TI, permitiendo tomar contacto con los procesos que siguen las empresas para conseguir maximizar sus beneficios.

Hoy en día Java sigue siendo uno de los lenguajes más populares en todo el mundo, con un uso muy extendido junto con otros lenguajes como C o .Net. Incluso las nuevas aplicaciones que se crean para sistemas Android se desarrollan mediante un Java modificado. En cuanto al área del Gobierno TI, ha tomado gran importancia en los últimos años, aunque tuvo sus orígenes en los años 80, fue en la primera década del año 2000 cuando comenzó a tener fuerza. Los datos que la aplicación utiliza están basados en el marco de trabajo COBIT que es una de las metodologías más importantes en el ámbito del Gobierno de las TI.

La aplicación por tanto puede ayudar realmente a empresas que deseen mejorar su rendimiento, conocer su madurez o tratar de resolver problemas TI, mediante el uso de una buena gestión del Gobierno TI proporcionada por el marco COBIT 4.1.

La herramienta final que se ha desarrollado no ha sido únicamente una tarea de programación sino que se han llevado a cabo todas las fases en el desarrollo de aplicaciones como son el análisis, el desarrollo e implementación, y ha sido documentada ampliamente con el fin de poder ser reutilizada o ampliada en el futuro.

Algunas de las ideas que se exponen en el apartado de “Líneas futuras” son interesantes para mejorar la aplicación, pero ya sin ellas, el programa estaría listo para usarse en el mundo empresarial y es por ello que finalmente ha dado como resultado una propiedad intelectual de la cual se ha hecho un registro de copyright para la Universidad Carlos III de Madrid.

Anexo A: Glosario

A

AI	Acquire and Implement (Adquirir e Implementar)
ANS	Acuerdo de Nivel de Servicio
API	Application Programming Interface (Interfaz de programación de aplicaciones)
ASCII	American Standard Code for Information Interchange (Código estándar estadounidense para el intercambio de información)

B

BSI	British Standards Institution
BSC	Balanced Scorecard (Cuadro de mando integral CMI)

C

CMI	Cuadro de Mando Integral
CMM	Capability Maturity Model (Modelo de Madurez de Capacidades)
CMMI	Capability Maturity Model Integration (Modelo de Integración y Capacidad de Madurez)
COBIT	Control Objectives for Information and related Technology (Objetivos de Control para la Información y las Tecnologías)
COCOMO	CONstructive COSt MODEL
COSO	Committee of Sponsoring Organizations of the Treadway Commission

D

DS	Deliver and Support (Entregar y Dar Soporte)
----	---

E

EDT	Estructura de Descomposición/Desglose de Trabajo
ER, ERM, ERD	(Modelo/Diagrama) Entidad Relación

(Entity Relationship (Model/Diagram))

G

GUI Graphic User Interface
(Interfaz gráfica de usuario)

H

HTML HyperText Markup Language
(Lenguaje de marcado de hipertexto)

I

IEC International Electrotechnical Commission
(Comisión Electrotécnica Internacional)

IP Internet Protocol

ISACA Information Systems Audit and Control Association
(Asociación de Auditoría y Control de Sistemas de Información)

ISO International Organization for Standardization
(Organización Internacional de Normalización)

IT Information Technology
(Tecnología de la Información)

ITGI IT Governance Institute
(Instituto de Gobierno TI)

ITIL Information Technology Infrastructure Library
(Biblioteca de Infraestructura de Tecnologías de Información)

J

JAR Java ARchive

J2EE Java 2 Enterprise Edition (edición empresarial)

J2SE Java 2 Standard Edition (edición estándar)

JDBC Java Database Connectivity
(Conectividad de base de datos Java)

JDK Java Development Kit
(Kit de desarrollo de Java)

JRE Java Runtime Environment
(Entorno en tiempo de ejecución de Java)

JVM Java Virtual Machine
(Máquina Virtual de Java)

L

LOPD	Ley Orgánica de Protección de Datos
LSSI	Ley de la Sociedad de los Servicios de Información

M

ME	Monitor and Evaluate (Monitorizar y Evaluar)
MOF	Microsoft Operation Framework (Marco de Referencia de Operaciones de Microsoft)
MVC	Modelo Vista Controlador

O

ORDBMS	Object-Relational Data Base Management System (Sistema de gestión de base de datos objeto-relacional)
--------	--

P

PDA	Personal Digital Assistant (Asistente Digital Personal)
PDF	Portable Document Format (Formato de documento portátil)
PHP	Hypertext Pre-processor
PO	Plan and Organise (Planificar y Organizar)
POO	Programación Orientada a Objetos

R

RACI	Responsible Accountable Consulted and Informed (Responsable Aprobador Consultado e Informado)
RAID	Redundant Arrays of Independent Disks (Conjunto redundante de discos independientes)
ROI	Return on Investment (Retorno de la inversión)

S

SDK	Software Development Kit
-----	--------------------------

	(Kit de desarrollo de software)
SEI	Software Engineering Institute
SGSI	Sistema de Gestión de la Seguridad de la Información
SLA	Service Level Agreement (Acuerdo de Nivel de Servicio, ANS)
SOX	ley Sarbanes-Oxley
SQL	Structured Query Language (Lenguaje de consulta estructurado)

T

TI	Tecnologías de la Información
TIC	Tecnologías de la Información y Comunicaciones
TSI	Tecnologías de los Sistemas de la Información y Comunicaciones

U

UCD	User-centered Design (Diseño centrado en el usuario)
UI	User Interface (Interfaz de Usuario)
UML	Unified Modeling Language (Lenguaje Unificado de Modelado)
URL	Uniform Resource Locator (Localizador uniforme de recursos)

W

WBS	Work Breakdown Structure (Estructura de Descomposición/Desglose de Trabajo EDT)
-----	--

Anexo B: Bibliografía

Auditoría Informática, Un enfoque práctico. 2ª Edición Ampliada y Revisada. Mario G. Piattini, Emilio del Peso. Ra-ma 2001.

Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación. Dolores Cuadra, Elena Castro, Ana Mª Iglesias, Paloma Martínez, Fco. Javier Calle, César de Pablo, Harith Al-Jumaily, Lourdes Moreno. Ra-ma 2007.

Diseño de Bases de Datos. Problemas Resueltos. Adoración de Miguel, Paloma Martínez, Elena Castro, José Mª Caverro, Dolores Cuadra, Ana Mª Iglesias, Carlos Nieto. Editorial Ra-ma 2000.

Fundamentos de Gestión de Servicios TI, basado en ITIL (2005). itSMF Internacional. Una publicación de itSMF International. Van Haren Publishing VHP. ITSM LIBRARY. The IT Service Management Forum.

Gobierno de las Tecnologías y los Sistemas de Información. Mario Piattini Velthius, Fernando Hervada Vidal. Ra-ma 2007.

Manual de Java. Osborne McGrawHill. Naughton.

Anexo C: Referencias

21 common problems within ITIL Incident Management Implementation.

<http://www.securityprocedure.com/21-common-problems-within-til-incident-management-implementation>

[En línea] 2 de 03 de 2012.

Altova. XML Editor, Data Management, UML, and Web Services Tools.

<http://www.altova.com/>

[En línea] 15 de 03 de 2012.

Aprenda Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales de San Sebastián. Universidad de Navarra. Javier García de Jalón, José Ignacio Rodríguez, Iñigo Mingo, Aitor Imaz, Alfonso Brazález, Alberto Larzabal, Jesús Calleja, Jon García. Febrero 2000.

<http://es.scribd.com/doc/56152925/Aprenda-Java-Como-Si-Estuviera-en-Primero-2000>

[En línea] 5 de 11 de 2011.

Best Price Computers. The Most Common Problems with PCs and their Solutions.

<http://www.bestpricecomputers.ltd.uk/freehelp/common.htm>

[En línea] 2 de 03 de 2012.

BSI España. Normas, Formacion, Auditoria y Certificacion.

<http://www.bsigroup.es/>

[En línea] 5 de 02 de 2012.

COBIT 4.1. IT Governance Institute (2007). Leading the IT Governance Community.

<http://www.isaca.org/Knowledge-Center/cobit/Pages/Downloads.aspx>

[En línea] 10 de 02 de 2012.

Dream in Code. Community Learning.

<http://www.dreamincode.net/>

[En línea] 18 de 03 de 2011.

Designing the user interface: text, colour, images, moving images and sound de OpenLearn LabSpace que está basado en guías de Götz (1998), Hartley (1994) and Rivlin et al. (1990). OpenLearn. LabSpace.

<http://labspace.open.ac.uk/>

[En línea] 16 de 02 de 2012.

Guía ISO 20000. Realizado por el Grupo de Trabajo “Difusión ISO 20000” del itSMF España. Septiembre 2008 - Versión 1. Autor: Juan José Carpintero. Co-Autores: Julio Ballesteros, Javier Benito, Manuel Díaz de la Fuente, Juan José Figueiras, Francisco Gil,

Orlando Pereda, Alejandro Pérez, Sergio Ribes, Marlon Molina. Dibujos: Juan José Carpintero.

<http://www.slideshare.net/albinogoncalves/respuestas-sobre-iso-20000>

[En línea] 23 de 06 de 2011.

GUI Look and Feel Standard desarrollado por DENR ITS. Está basado en Microsoft's Web site (<http://www.microsoft.com/win32dev/uiguide/default.htm>); "The Windows Interface Guidelines for Software Design" published by Microsoft Press; "GUI Design Essentials" by Weinschenk, Jamar and Yeo. N.C.DENR. N.C Department of Environment and Natural Resources.

http://portal.ncdenr.org/c/document_library/get_file?uuid=5ad40d11-2686-442b-9cd0-462e3e3e4524&groupId=17979

[En línea] 16 de 02 de 2012.

Isaca.

<http://www.isaca.org>

[En línea] 10 de 03 de 2012.

Isaca Madrid.

<http://www.isacamadrid.es>

[En línea] 15 de 03 de 2012.

ISO/IEC 20000 el estándar para la Gestión de Servicios TI. Alejandro M. Pérez Sánchez. Gerente Dirección de Tecnología y Sistemas de Telefónica Gestión de Servicios Compartidos. Secretario WG25-AENOR-ISO/EC "Gestión y Buen Gobierno de TI". Socio fundador itSMF España. Socio senior de ATI. Email: alejandro.perezsanchez@telefonica.es.

<http://www.slideshare.net/albinogoncalves/iso-20000-el-estndar-para-la-gestin-de-servicios-de-ti>

[En línea] 14 de 10 de 2011.

IT Compliance. Un reto de obligado cumplimiento. José Luis Sánchez.

DIRECTOR GENERAL AnyHelp International. Tribuna de opinión.

<http://www.revista-ays.com/DocsNum25/Tribuna/Sanchez.pdf>

[En línea] 3 de 01 de 2012.

iText. Free / Open Source PDF Library for Java and C#.

<http://itextpdf.com/>

[En línea] 16 de 02 de 2012.

IT Governance Institute.

<http://www.itgi.org>

[En línea] 22 de 01 de 2012.

IT Governance Network.

<http://itgovernance.com>

[En línea] 9 de 03 de 2012.

itSMF España.

<http://www.itsmf.es/>

[En línea] 22 de 01 de 2012.

itSMF International.

<http://www.itsmfi.org/>

[En línea] 22 de 01 de 2012.

Layouts Managers: Introducción a los layouts managers. Martín Pérez Mariñán.

<http://usuarios.multimania.es/manualesjava/manuales/layouts/layouts.pdf>

[En línea] 2 de 05 de 2011.

Metodologías y Estándares: ITIL, CMMI, ISO9000, COSO, COBIT y Six Sigma.

<http://www.nohaylimites.com/?p=243/>

[En línea] 08 de 02 de 2012.

Microsoft Learning.

<http://www.microsoft.com/spain/formacion>

[En línea] 5 de 02 de 2012.

MySQL Workbench.

<http://dev.mysql.com/downloads/workbench/>

[En línea] 2 de 03 de 2012.

Ofuscación de código.

<http://www.iec.csic.es/criptonomicon/java/ofuscacion.html>

[En línea] 2 de 03 de 2012.

Oracle Consulting, Oracle Support and Oracle Training by BC Oracle Consulting.
Burleson Consulting.

<http://www.dba-oracle.com/>

[En línea] 8 de 03 de 2012.

Oracle JDBC Drivers.

<http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-10201-088211.html>

[En línea] 15 de 04 de 2011.

Stack Overflow.

<http://stackoverflow.com/>

[En línea] 13 de 12 de 2011.

Swing y JFC (Java Foundation Classes) Autor: Sun. Traductor: Juan Antonio Palos (Ozito).

<http://es.scribd.com/trukly/d/7545519-Manual-JAVA-Swing>

[En línea] 20 de 08 de 2011.

The Java Tutorials.

<http://download.oracle.com/javase/tutorial/>

[En línea] 15 de 03 de 2012.

The Windows Interface Guidelines — A Guide for Designing Software. Microsoft Windows. February 1995.

http://www.ics.uci.edu/~kobsa/courses/ICS104/course-notes/Microsoft_WindowsGuidelines.pdf

[En línea] 16 de 02 de 2012.

Tutoriales y ejemplos de Java.

<http://www.chuidiang.com/>

[En línea] 12 de 03 de 2012.

Universidad Politécnica de Madrid. Facultad de Informática. Estudios de Doctorado. DESARROLLO DE PROCESOS DE GESTIÓN DE SERVICIOS DE EXPLOTACIÓN SIGUIENDO EL MODELO CMMI. Alumno: Sr. Juan Raggio Pérez, jraggio@endesa.es Tutor: Sr. Tomás San Feliu Gilabert, tsanfe@fi.upm.es.

<http://is.ls.fi.upm.es/doctorado/Trabajos20042005/Raggio.pdf>

[En línea] 21 de 09 de 2011.

Wikipedia, la enciclopedia libre.

<http://es.wikipedia.org>

[En línea] 13 de 03 de 2012.

Wikipedia, the free encyclopedia.

<http://en.wikipedia.org>

[En línea] 10 de 03 de 2012.

Anexo D: Manual del usuario

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA PARA LA RESOLUCIÓN DE PROBLEMAS EN EL ÁMBITO DEL GOBIERNO DE LAS TI BASADO EN EL ESTÁNDAR COBIT 4.1

Manual del usuario



**UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**

**Vinculado con un Proyecto Fin de Carrera de
Ingeniería Técnica en Informática de Gestión**

Autores:

Juan Manuel Acedo Ruiz
Jorge Bustos Iglesias

Tutora:

Anabel Fraga Vázquez

Índice de contenidos

Introducción	4
Requisitos de funcionamiento.....	5
Instalación	6
Ficheros de configuración	8
Tipos de usuarios	13
Interfaz gráfica y primeros pasos.....	17
La pantalla principal	19
El perfil usuario.....	24
Búsqueda de problemas y solución de los mismos mediante la metodología <i>COBIT 4.1</i>	25
Realización de test de evaluación.....	30
Consulta de procesos <i>COBIT 4.1</i> a través de dominios <i>COBIT 4.1</i>	33
Opciones de la interfaz gráfica:	36
Modificación del idioma:	36
Modificación de la apariencia de la interfaz gráfica:	37
Ayuda:	39
Manual del usuario:	39
Información sobre el programa:	39
El perfil administrador	40
Opciones de la BBDD:.....	43
Modificación de la contraseña del usuario administrador:.....	44
Modificación de la información relacionada con los problemas de la BBDD:.....	46
Añadir información:	46
Eliminar información:	47
Actualizar información:	48
Modificación de la información relacionada con los tests de evaluación de la BBDD:.....	51
Añadir información:	51
Eliminar información:	52
Actualizar información:	53
Modificación de la información relacionada con <i>COBIT 4.1</i> :.....	55
Añadir información:	55

Eliminar información:	56
Actualizar información:	57
Resolución de problemas	59
Copyright	59

Introducción

En primer lugar le damos la bienvenida al software **IT Governance Solver v. 1.0**. Con esta aplicación podrá gestionar y resolver todas las cuestiones referentes al Gobierno de TI mediante una interfaz sencilla e intuitiva, que tiene como base la metodología **COBIT 4.1**.

Mediante este manual se pretende que el uso de este software sea lo más efectivo posible, además de servir como guía para todas aquellas cuestiones que se puedan plantear durante su utilización.

Requisitos de funcionamiento

Para que **IT Governance Solver v. 1.0** pueda funcionar correctamente, es necesario el siguiente entorno de ejecución:

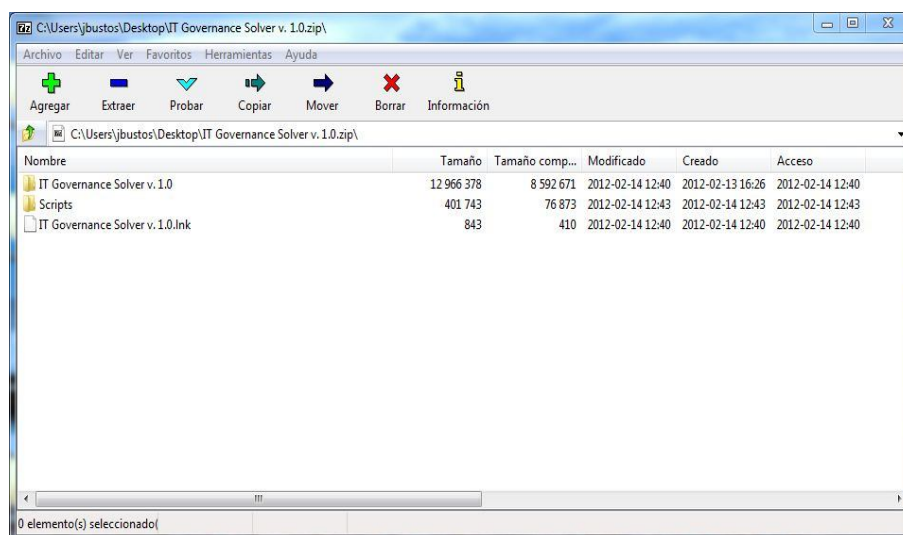
- **PC compatible con Microsoft Windows.**
- **Sistemas Operativos soportados:**
 - Microsoft Windows XP - 32 bits.
(En cualquiera de sus versiones disponibles).
 - Microsoft Windows XP - 64 bits.
(En cualquiera de sus versiones disponibles).
 - Microsoft Windows Vista - 32 bits.
(En cualquiera de sus versiones disponibles).
 - Microsoft Windows Vista - 64 bits.
(En cualquiera de sus versiones disponibles).
 - Microsoft Windows 7 - 32 bits.
(En cualquiera de sus versiones disponibles).
 - Microsoft Windows 7 - 64 bits.
(En cualquiera de sus versiones disponibles).
- **Bases de Datos soportadas:**
 - Oracle Database 10g – con SQL Plus.
(En cualquiera de sus versiones disponibles).
 - Oracle Database 11g – con SQL Plus.
(En cualquiera de sus versiones disponibles).
- **Java Runtime Environment:**
 - JAVA JRE.
(En cualquiera de sus versiones disponibles).
- **Compresor/Descompresor de ficheros en formato Zip.**
- **Visor de documentos en PDF.**
- **Visor de texto plano.**
- **Navegador de Internet compatible con HTML.**
(En cualquiera de sus versiones disponibles).

Se permite la instalación y el uso de cualquier otro software siempre y cuando no interfiera en el funcionamiento correcto del sistema operativo, de la base de datos, y del entorno de ejecución de JAVA.

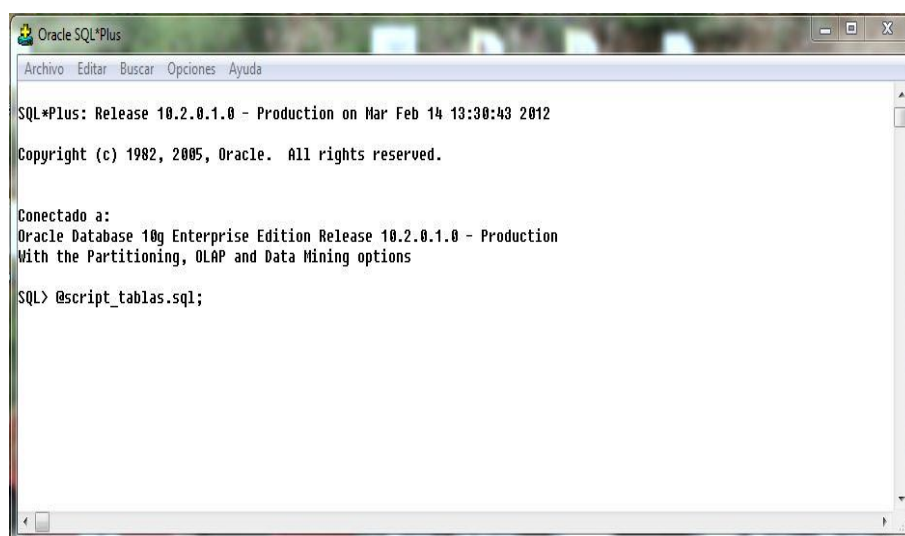
Instalación

La instalación de **IT Governance Solver v. 1.0** se realiza de forma muy sencilla, y consta de dos pasos, que deben ser realizados una vez que el sistema dispone de los requisitos de funcionamiento:

1. Descompresión del contenido del fichero comprimido **IT Governance Solver v. 1.0.zip** a una ubicación cualquiera desde donde se quiera ejecutar el programa. Se recomienda descomprimir el fichero en el Escritorio de Windows, a fin de que sea más sencilla su ejecución. Dicho fichero comprimido contiene un acceso directo llamado **IT Governance Solver v. 1.0**, una carpeta llamada **IT Governance Solver v. 1.0**, y por último otra carpeta con el nombre **Scripts**.



2. Una vez descomprimido todo el contenido, hay que abrir la carpeta Scripts, e introducir en la BBDD del sistema –por ejemplo mediante SQL Plus–, los scripts de la BBDD de la siguiente forma:
 - a. Primero se introduce el script ***script_tablas.sql***, que es el encargado de borrar las tablas del programa en la BBDD en el caso de que ya existieran, y de crearlas para posteriormente introducir los datos iniciales del programa.
 - b. Por último, se introduce el ***script_insert.sql***, que contiene todos los datos iniciales del programa, incluida la metodología **COBIT 4.1**.

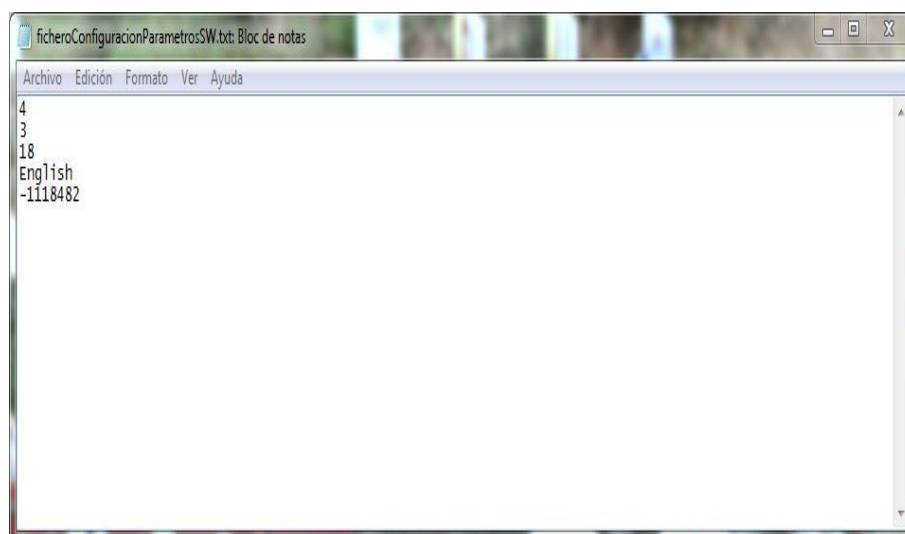


Una vez realizados estos pasos el programa queda totalmente instalado en el sistema.

Ficheros de configuración

Una vez instalado **IT Governance Solver v. 1.0**, la configuración de los parámetros del programa se encuentra almacenada en la carpeta ficheros, dentro del directorio raíz de **IT Governance Solver v. 1.0**, y su contenido es el siguiente:

- **Fichero `ficheroConfiguracionParametrosSW.txt`:** Este fichero incluye la configuración de los parámetros de la interfaz del programa que se pueden modificar. Se puede configurar también desde la interfaz del programa. A continuación el significado de cada línea:

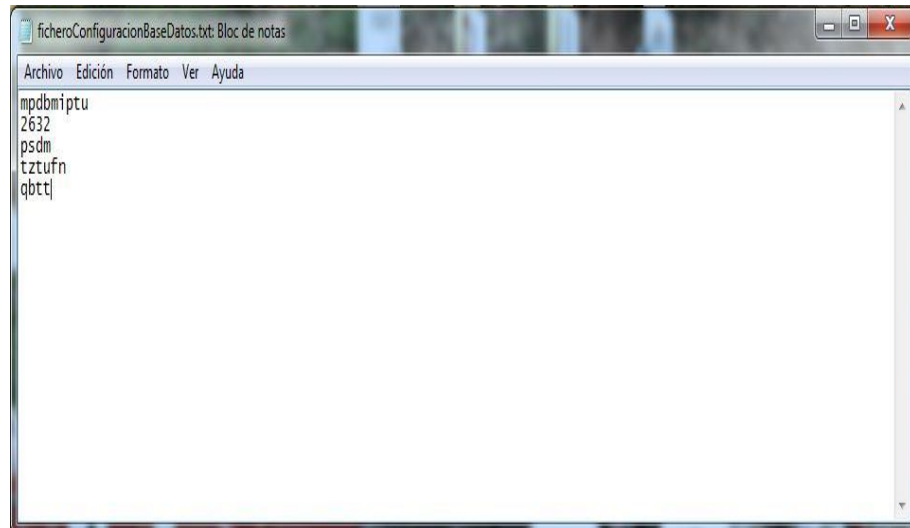


- **Primera línea:** Contiene el número de problemas que se muestran tras una búsqueda en el modo usuario del programa.
- **Segunda línea:** Contiene el número de assessments que se muestran en una ventana en el modo usuario.
- **Tercera línea:** Contiene el tamaño de la letra de la barra de menús del programa, en puntos.
- **Cuarta línea:** Contiene el idioma de toda la interfaz del programa. English para inglés, y Spanish para español. El lenguaje español no está implementado de forma completa.
- **Quinta línea:** Contiene el código del color de la barra de menús del programa, en formato RGB.

IMPORTANTE: Si el fichero **ficheroConfiguracionParametrosSW.txt** no está creado inicialmente, el programa lo crea con unos valores por defecto que a continuación se detallan:

- **Primera línea:** 4
- **Segunda línea:** 3
- **Tercera línea:** 18
- **Cuarta línea:** English
- **Quinta línea:** -1118482

- **Fichero *ficheroConfiguracionBaseDatos.txt*:** Este fichero incluye la configuración de la base de datos de Oracle con la que se conecta el programa. El contenido del mismo está cifrado por motivos de seguridad y sólo se debe modificar desde la administración de **IT Governance Solver v. 1.0** –que se detallará posteriormente-. No obstante se detalla a continuación el significado de cada línea:

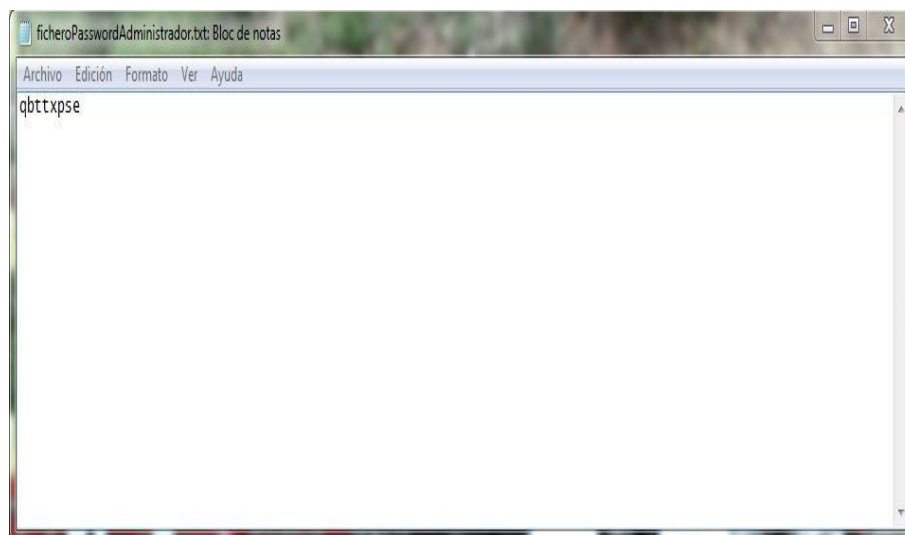


- **Primera línea:** Contiene el host al que se conecta el programa. Es decir, la dirección lógica donde está almacenada la base de datos.
- **Segunda línea:** Contiene el puerto de la base de datos a la que se conecta el programa.
- **Tercera línea:** Contiene el nombre de la cadena de la base de datos a la que se conecta el programa.
- **Cuarta línea:** Contiene el nombre de usuario de la base de datos a la que se conecta el programa.
- **Quinta línea:** Contiene la contraseña de la base de datos a la que se conecta el programa.

IMPORTANTE: Si el fichero **ficheroConfiguracionBaseDatos.txt** no está creado inicialmente, el programa lo crea con unos valores por defecto que a continuación se detallan:

- **Primera línea:** localhost
(Contenido cifrado).
- **Segunda línea:** 1521
(Contenido cifrado).
- **Tercera línea:** orcl
(Contenido cifrado).
- **Cuarta línea:** system
(Contenido cifrado).
- **Quinta línea:** pass
(Contenido cifrado).

- **Fichero *ficheroPasswordAdministrador.txt*:** Este fichero incluye contraseña del usuario administrador del programa. El contenido del mismo está cifrado por motivos de seguridad y sólo se debe modificar desde la administración de **IT Governance Solver v. 1.0** –que se detallará posteriormente-. No obstante se detalla a continuación el significado de cada línea:



- o **Primera línea:** Contiene la contraseña del usuario administrador del programa.

IMPORTANTE: Si el fichero ***ficheroPasswordAdministrador.txt*** no está creado inicialmente, el programa lo crea con un valor por defecto que a continuación se detalla:

- o **Primera línea:** password
(Contenido cifrado).

Tipos de usuarios

IT Governance Solver v. 1.0, tiene dos tipos de usuarios con perfiles bien diferenciados:

- **Usuario:** Este tipo de perfil de usuario permite la interacción con el programa sin la modificación de los datos contenidos en la BBDD. Sí se pueden cambiar las preferencias visuales de la interfaz del programa. En concreto tiene acceso a las siguientes funciones:
 - **Metodología COBIT 4.1:**
 - Búsqueda de problemas y solución de los mismos mediante la metodología **COBIT 4.1**.
 - Realización de tests de evaluación.
 - Consulta de procesos **COBIT 4.1** a través de dominios **COBIT 4.1**.
 - **Opciones de la interfaz gráfica:**
(Contenidas en el fichero *ConfiguracionParametrosSW.txt*).
 - Modificación del idioma de la interfaz gráfica del programa – disponible en inglés y en español (este último no está implementado de forma completa en esta versión)-.
 - Modificación del número de problemas mostrados por ventana al realizar la búsqueda de problemas.
 - Modificación del número de evaluaciones mostradas por ventana al realizar un test de evaluación.
 - Modificación del tamaño de las letras de la barra de menú superior –en puntos-.
 - Modificación del color de la barra de menú superior –en formato RGB-.
 - Aplicar los valores por defecto.
 - **Guardar una solución:**
 - En formato txt, PDF, y HTML.
 - **Impresión de una solución.**
 - **Ayuda:**
 - Manual del usuario en idioma español o inglés (no disponible este último en esta versión) –en formato PDF-.
 - Información sobre el programa.

- **Administrador:** Este tipo de perfil de usuario permite la interacción con el programa y la modificación de los datos contenidos en la Base de Datos (BBDD), así como de dos de los tres ficheros de configuración que maneja el software. Se requiere de una contraseña para entrar en el modo administrador. En concreto el perfil tiene acceso a las siguientes funciones, además de las del perfil *Usuario*:
 - **Opciones de la BBDD:**
(Contenidas y cifradas en el fichero *ConfiguracionBaseDatos.txt*).
 - Modificación del host de la BBDD.
 - Modificación del puerto de la BBDD.
 - Modificación de la cadena de la BBDD.
 - Modificación del usuario de la BBDD.
 - Modificación de la contraseña de la BBDD.
 - **Modificación de la contraseña del usuario administrador.**
(Contenida y cifrada en el fichero *PasswordAdministrador.txt*).
 - **Modificación de la información relacionada con los problemas de la BBDD:**
 - Añadir información:
 - Añadir problemas.
 - Añadir palabras.
 - Añadir sinónimos.
 - Eliminar información:
 - Eliminar problemas.
 - Eliminar palabras.
 - Eliminar sinónimos.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los problemas.
 - Modificar los nombres de las palabras.
 - Modificar los nombres de los sinónimos.
 - Relacionar información:
 - Relacionar/des-relacionar problemas con procesos **COBIT 4.1**.
 - Relacionar/des-relacionar palabras con problemas.
 - Relacionar/des-relacionar palabras con sinónimos.

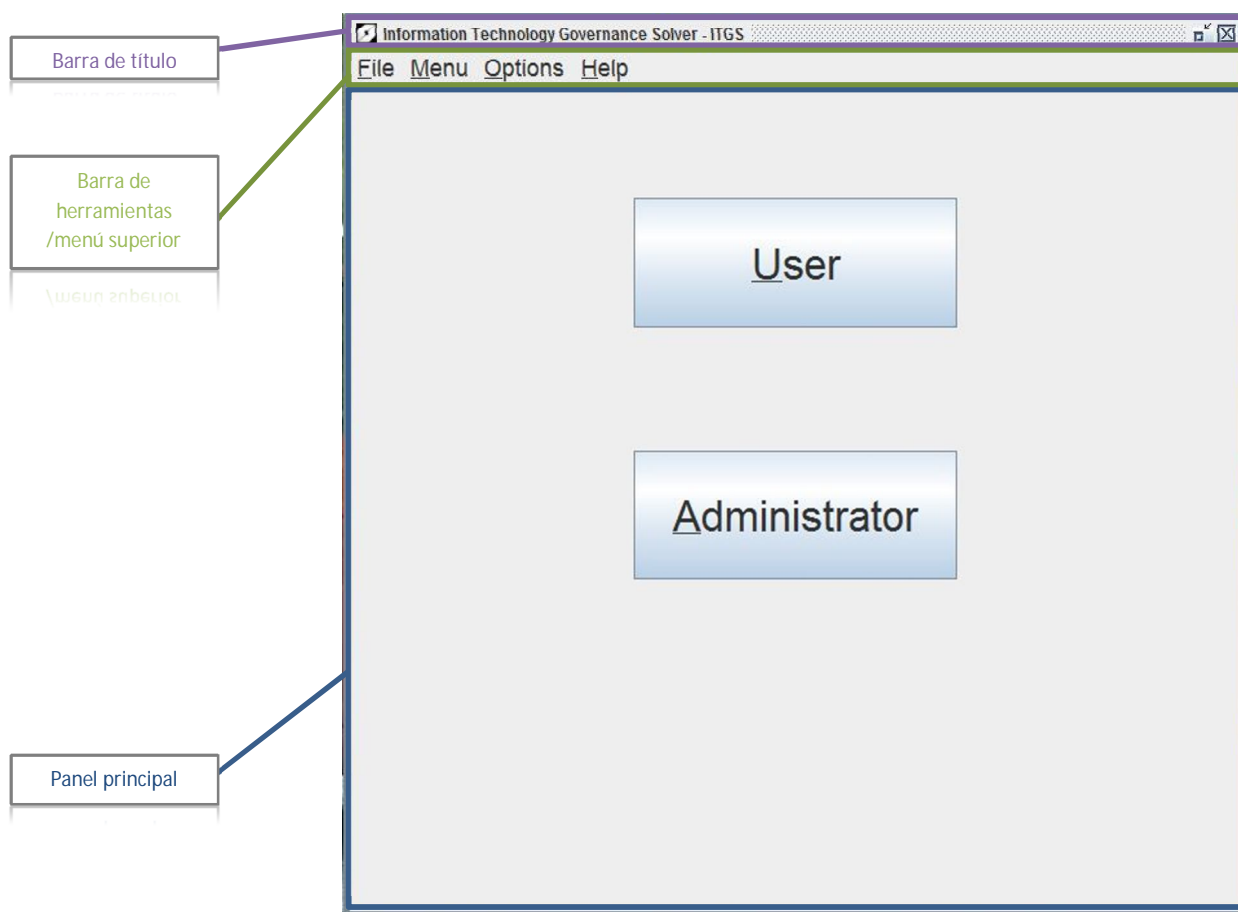
- **Modificación de la información relacionada con los tests de evaluación de la BBDD:**
 - Añadir información:
 - Añadir test de evaluación.
 - Eliminar información:
 - Eliminar test de evaluación.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los tests de evaluación.
 - Relacionar información:
 - Relacionar/des-relacionar tests de evaluación con procesos **COBIT 4.1**.
- **Modificación de la información relacionada con COBIT 4.1:**
 - Añadir información:
 - Añadir dominios.
 - Añadir procesos.
 - Añadir objetivos de control.
 - Añadir objetivos TI.
 - Añadir objetivos de negocio.
 - Añadir metas.
 - Añadir métricas.
 - Añadir acciones.
 - Añadir modelos de madurez.
 - Eliminar información:
 - Eliminar dominios.
 - Eliminar procesos.
 - Eliminar objetivos de control.
 - Eliminar objetivos TI.
 - Eliminar objetivos de negocio.
 - Eliminar metas.
 - Eliminar métricas.
 - Eliminar acciones.
 - Eliminar modelos de madurez.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los dominios.
 - Modificar los nombres de los procesos.
 - Modificar los nombres de los objetivos de control.
 - Modificar los nombres de los objetivos TI.
 - Modificar los nombres de los objetivos de negocio.
 - Modificar los nombres de los metas.
 - Modificar los nombres de las métricas.
 - Modificar los nombres de los acciones.

- Modificar los nombres de los modelos de madurez.
- Relacionar información:
 - Relacionar/des-relacionar dominios con procesos.
 - Relacionar/des-relacionar procesos con objetivos de control.
 - Relacionar/des-relacionar objetivos de control con objetivos TI.
 - Relacionar/des-relacionar objetivos TI con objetivos de negocio.
 - Relacionar/des-relacionar metas con métricas.
 - Relacionar/des-relacionar métricas con objetivos de control.
 - Relacionar/des-relacionar acciones con objetivos de control.
 - Relacionar/des-relacionar modelos de madurez con procesos.

Interfaz gráfica y primeros pasos

Una vez instalado **IT Governance Solver v. 1.0**, basta hacer doble clic en el acceso directo del programa (**IT Governance Solver v. 1.0**) para ejecutarlo. Llegados a este punto, se mostrará la pantalla principal.

La interfaz gráfica de **IT Governance Solver v. 1.0** está dividida de la siguiente forma:



Como se puede apreciar en la ilustración anterior, la interfaz gráfica consta de los siguientes elementos:

- **Barra de título** con el nombre del programa, así como los botones de minimizar y de cerrar.
- **Barra de herramientas superior** que contiene el acceso directo a las funcionalidades principales del programa.
 - o **Menú File:**
 - Save as...: Permite guardar una solución en los formatos TXT, HTML, y PDF.
 - Print: Permite imprimir como demostración, la primera página de una solución.
 - Exit: Cierra el programa.

- **Menú Menu:**
 - Main: Muestra la pantalla principal del programa.
 - User: Permite el acceso a las diferentes funcionalidades del perfil usuario.
 - Administrator: Permite el acceso previo login con password, a las diferentes funcionalidades del perfil administrador.
 - **Menú Options:**
 - Language: Permite el cambio de idioma de la interfaz gráfica del programa a inglés o español, no estando este último implementado de forma completa en esta versión.
 - Preferences: Permite la modificación de los siguientes parámetros de la interfaz gráfica:
 - Modificación del número de problemas mostrados al realizar la búsqueda de problemas.
 - Modificación del número de evaluaciones mostradas al realizar un test de evaluación.
 - Modificación del tamaño de las letras de la barra de menú superior –en píxeles-.
 - Modificación del color de la barra de menú superior –en formato RGB-.
 - **Menú Help:**
 - User guide: Manual del usuario en idioma inglés –en formato PDF- y no disponible en esta versión.
 - About: Información sobre el programa.
- **Un panel principal** en el que se mostrará el contenido del programa, a excepción de la solución que será visualizada en una ventana aparte.

La pantalla principal

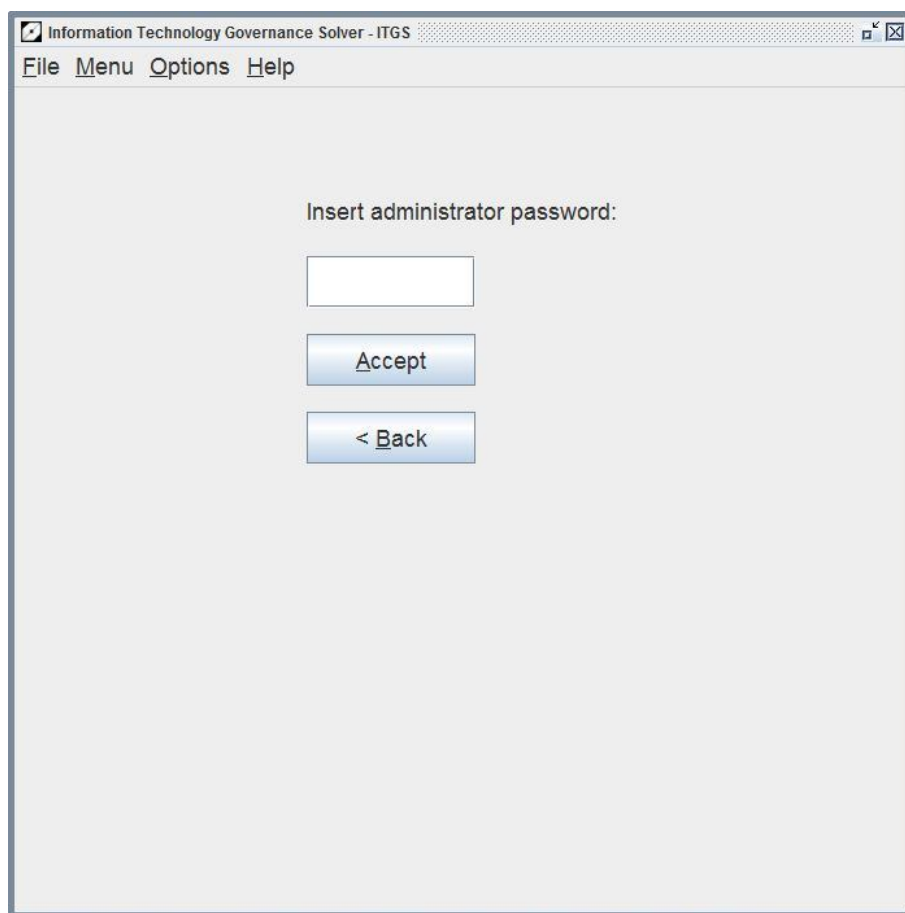
La pantalla principal es la primera pantalla que se muestra tras ejecutar el programa, y permite el acceso a todas las funcionalidades del programa.

Para utilizar el programa correctamente, primero se deben introducir los parámetros correctos de conexión de la base de datos de Oracle a la que se conectará. De lo contrario, el programa no funcionará correctamente y se mostrará un error de conexión a la base de datos:



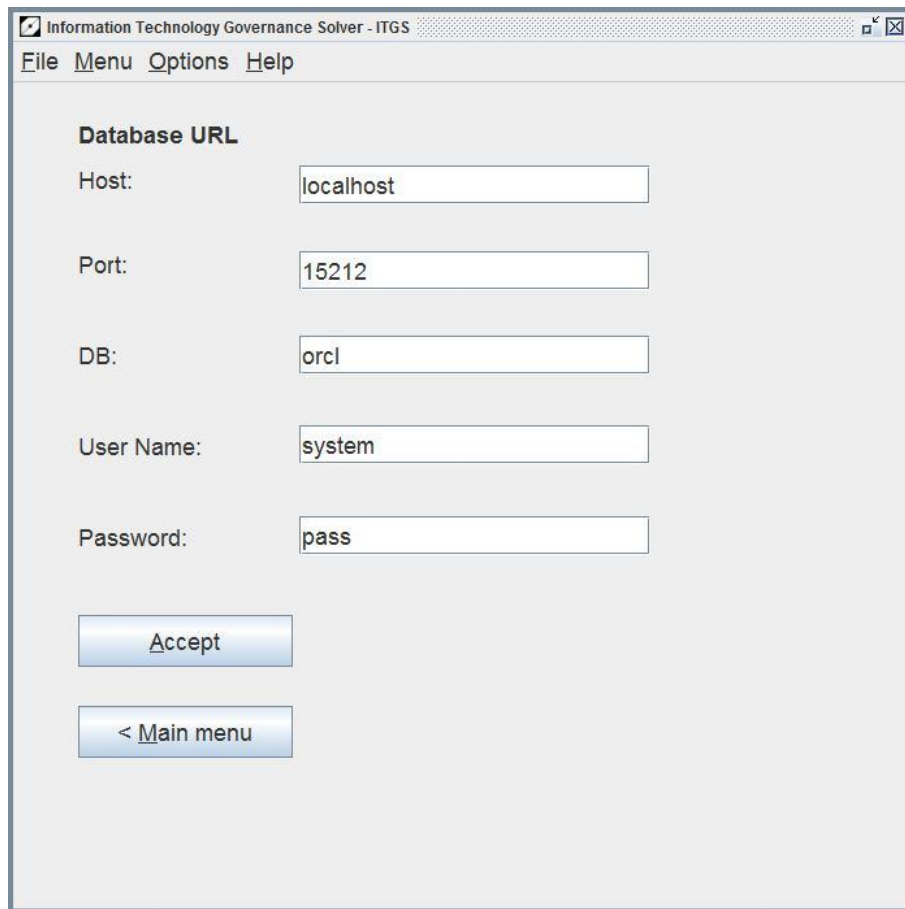
Salvo en el caso de que los parámetros por defecto del fichero de configuración de la Base de Datos sean exactamente los correctos, habrá que introducirlos la primera vez que se quiera utilizar el programa, o si se realiza cualquier cambio en la configuración del acceso a la base de datos. Para ello, desde la ventana principal del programa, hacemos clic en el botón **"Administrator"**.

Se mostrará una nueva ventana que permite el cambio:



Como es la primera vez que se ejecuta el programa, como password hay que teclear *password* en el cuadro de texto, y a continuación hacer clic en el botón "**Accept**".

Se mostrará la siguiente ventana, donde se podrá configurar el acceso a la Base de Datos de Oracle con la que se conecta el programa:

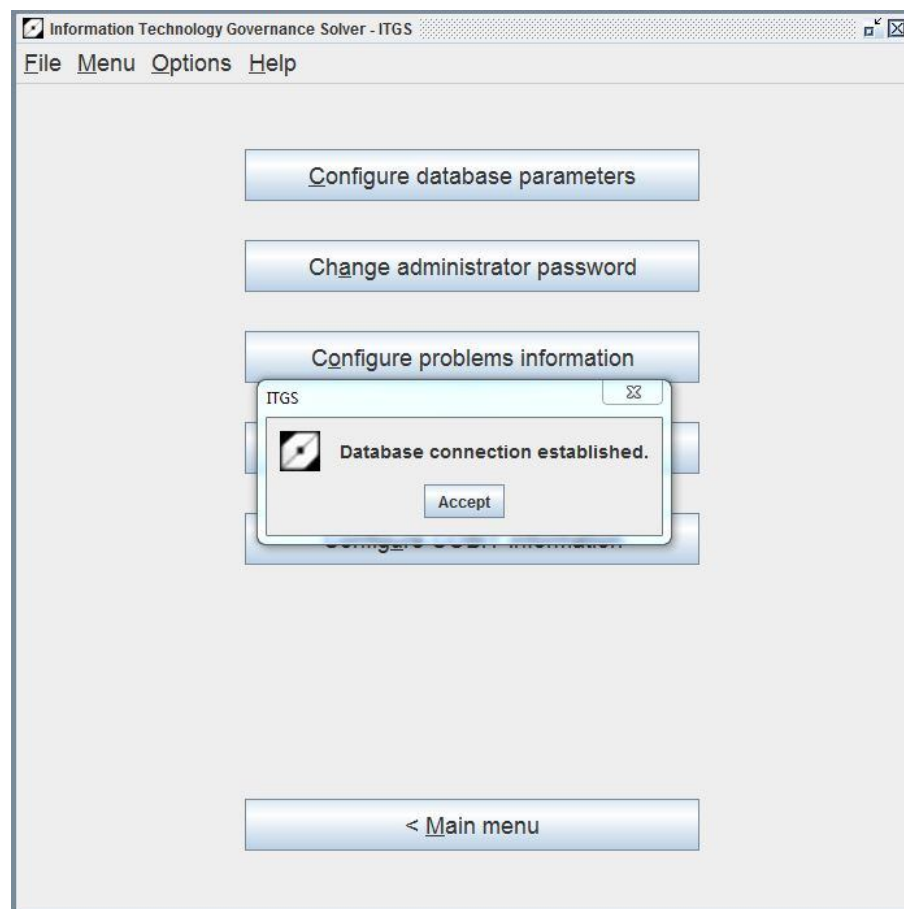


The screenshot shows a window titled "Information Technology Governance Solver - ITGS". It has a menu bar with "File", "Menu", "Options", and "Help". Below the menu bar, there is a section titled "Database URL". This section contains five input fields: "Host:" with the value "localhost", "Port:" with the value "15212", "DB:" with the value "orcl", "User Name:" with the value "system", and "Password:" with the value "pass". At the bottom of the form, there are two buttons: "Accept" and "< Main menu".

Explicación de los campos a insertar:

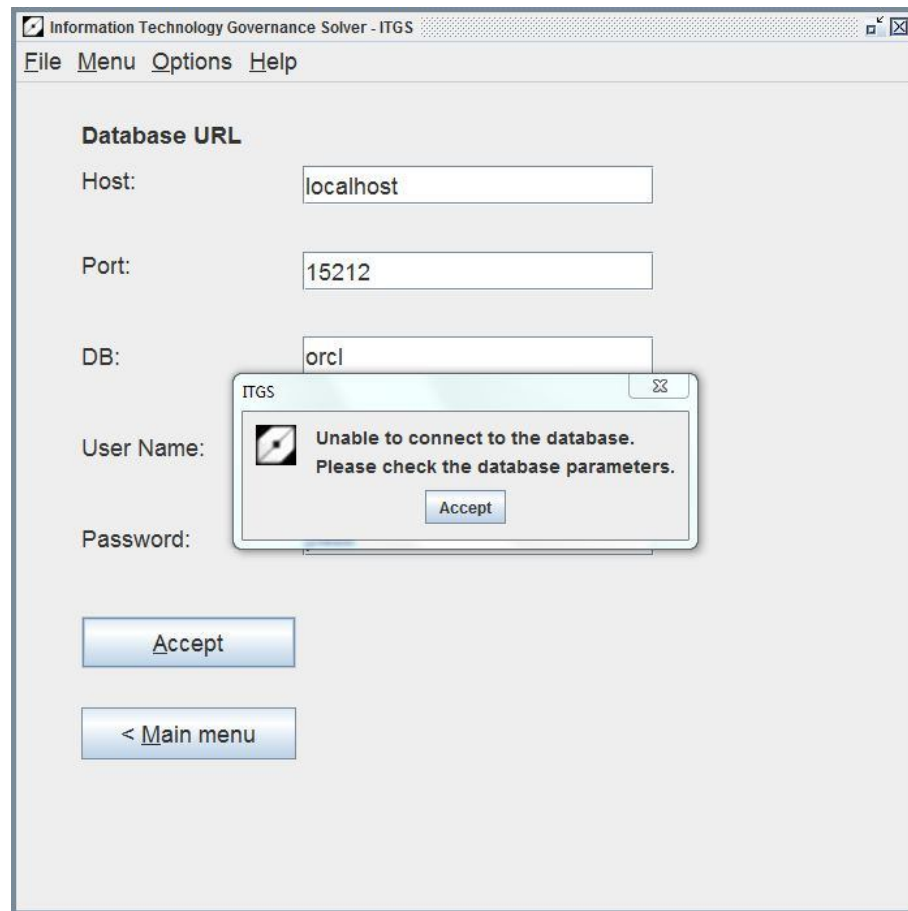
- **Host:** Hace referencia a la ubicación donde está instalada la base de datos. Por ejemplo, si está instalada en el equipo desde el que se utiliza el programa, su valor será *localhost*. En otro caso, se debe rellenar con la dirección IP o el nombre lógico del equipo donde esté instalada la base de datos.
- **Port:** Hace referencia al puerto desde el cual se accede a la base de datos.
- **DB:** Indica el nombre de la base de datos a la que se conectará el programa.
- **User Name:** Indica el nombre de usuario de la base de datos.
- **Password:** Indica la contraseña de la base de datos.

Una vez insertados todos los valores de forma correcta, se mostrará una ventana indicando que se ha podido establecer la conexión:



Hacemos clic en "**Accept**" y ya está el programa preparado para funcionar de forma correcta.

En caso contrario, mediante otra ventana se indicará que no se ha podido establecer la conexión, y que hay que volver a insertar los campos de nuevo:



El perfil usuario

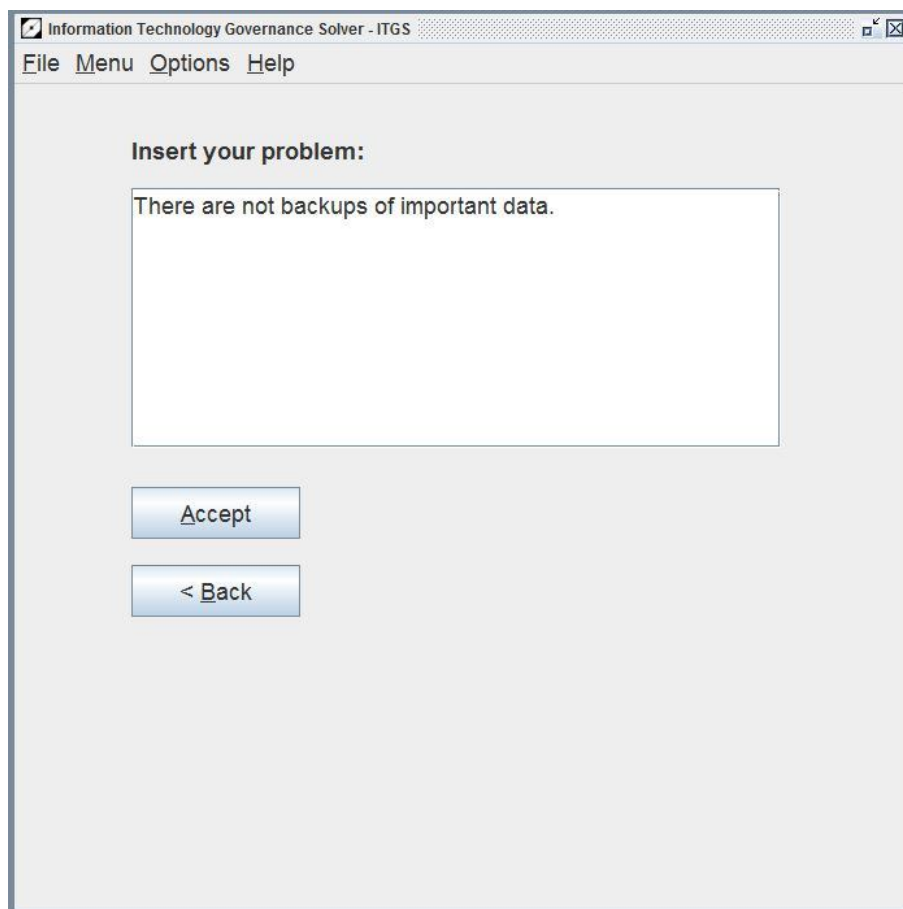
Desde el perfil usuario del programa se pueden realizar las siguientes operaciones:

- **Metodología COBIT 4.1:**
 - Búsqueda de problemas y solución de los mismos mediante la metodología **COBIT 4.1**.
 - Realización de tests de evaluación.
 - Consulta de procesos **COBIT 4.1** a través de dominios **COBIT 4.1**.

Búsqueda de problemas y solución de los mismos mediante la metodología COBIT 4.1

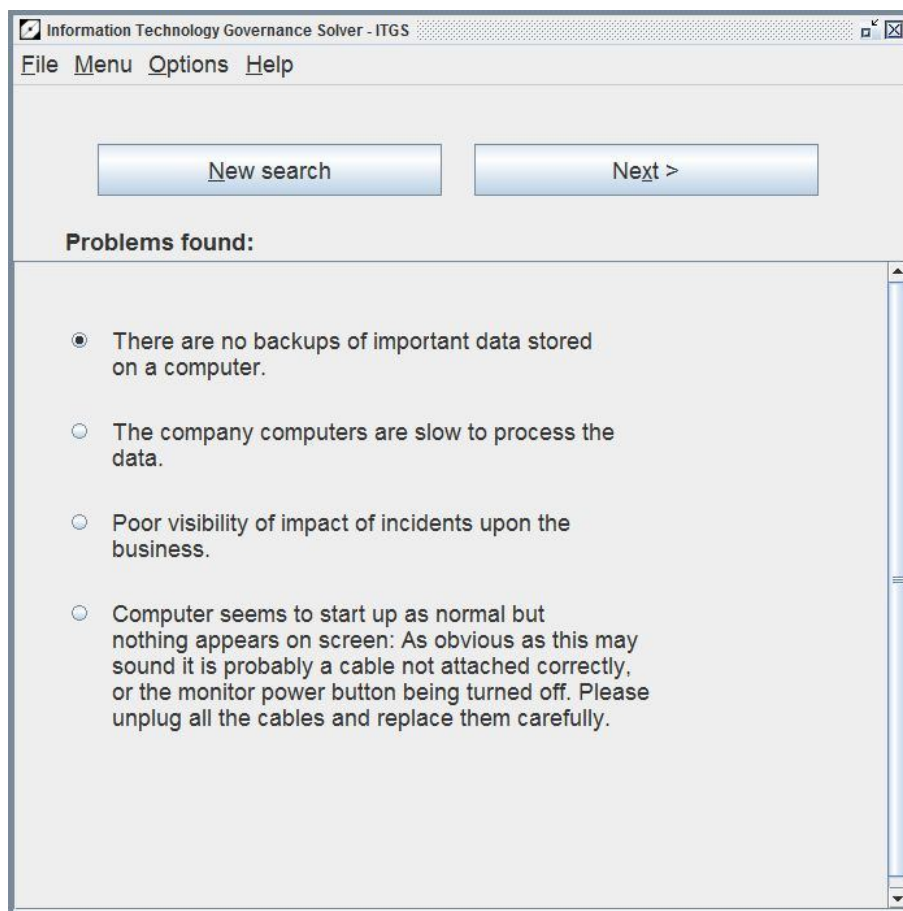
Para buscar un determinado problema o varios, hay que seguir los siguientes pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**User**".
- 2º Hacer clic en el botón "**Look up problems**".
- 3º Introducir el problema a buscar en el cuadro de texto.

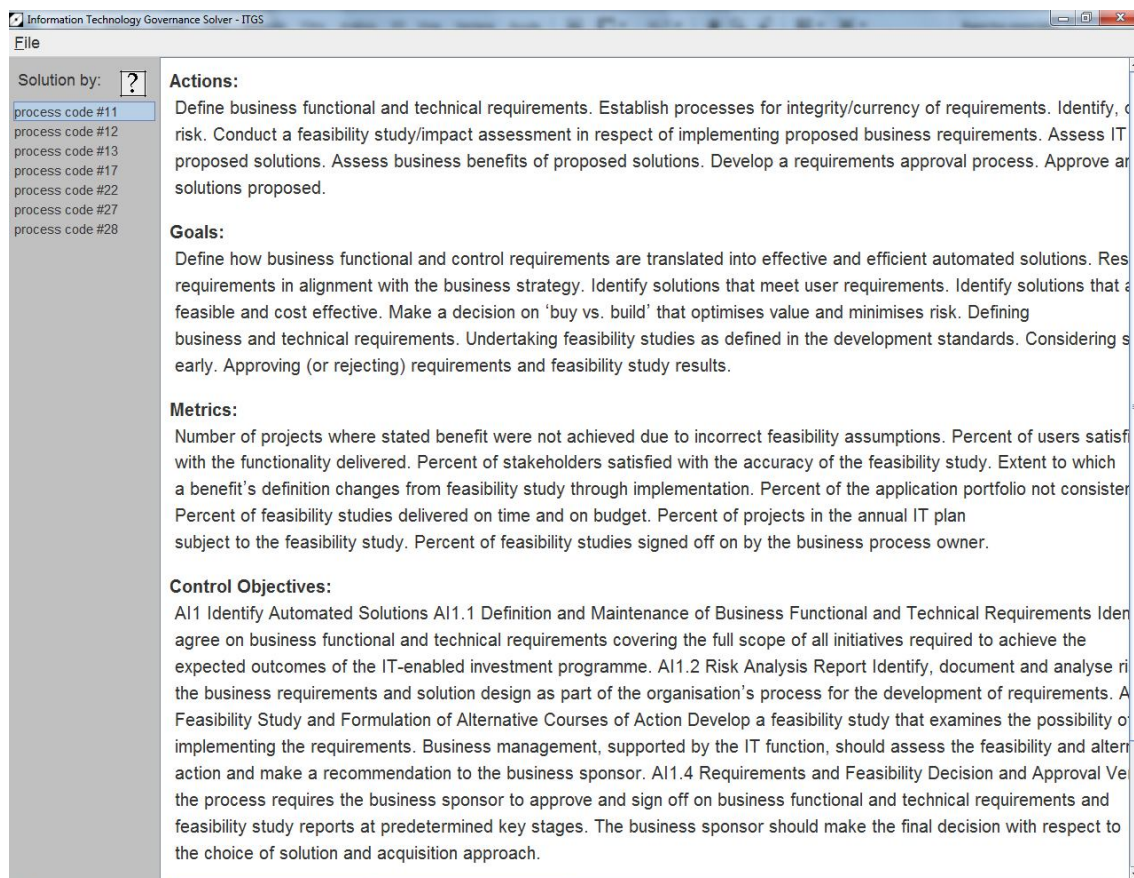


- 4º Hacer clic en el botón "**Accept**".

5º Si hay problemas encontrados se mostrará un listado de problemas, pudiendo elegir uno de ellos para que se muestre su solución en una nueva ventana, o la posibilidad de realizar una nueva búsqueda. Una vez elegido el problema, hacer clic en el botón "**Next >**".



6º Aparece una nueva ventana, en la que se muestra la solución del problema siempre que exista, basada en los procesos de **COBIT 4.1** que se han de realizar para resolverlo.



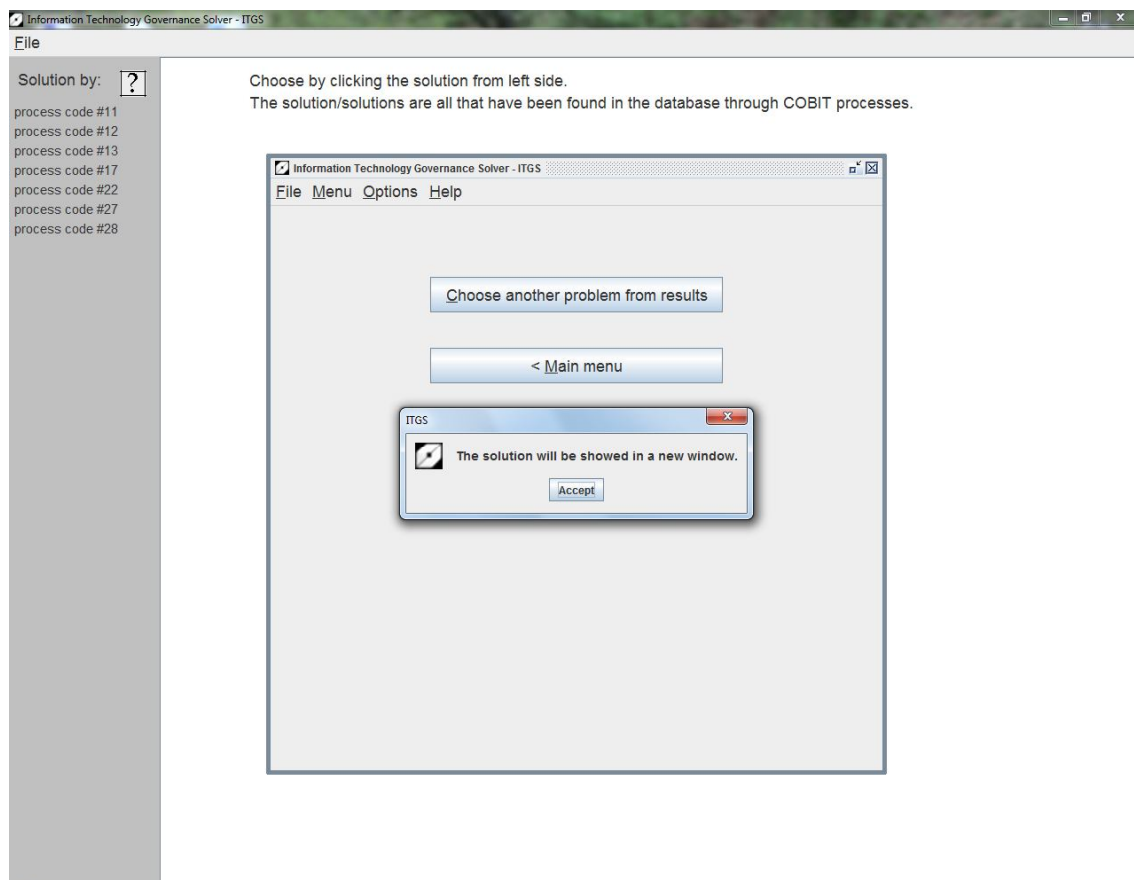
Además, mediante el menú **"File"** situado en la parte superior de la ventana de la solución, ésta puede ser almacenada en un fichero de texto plano (txt), en un fichero en formato HTML, o en un fichero en formato PDF.

El programa ofrece también la posibilidad de imprimir como demostración, la primera página de la solución que se muestra por pantalla, también mediante el menú **"File"** situado en la parte superior de la ventana de la solución.

Haciendo clic en el icono de interrogación de la parte izquierda de esta ventana, el usuario podrá conocer a qué proceso de **COBIT 4.1** pertenecen los códigos que se muestran de los procesos.



7º Además, el programa ofrece la posibilidad de seleccionar otros problemas del listado de problemas encontrados anteriormente tras la introducción del problema por parte del usuario, o de navegar hasta la pantalla principal del programa.



Realización de test de evaluación

Para realizar los tests de evaluación, hay que seguir los siguientes pasos:

1º En la pantalla principal del programa, hacer clic en el botón "**User**".

2º Hacer clic en el botón "**Do assessments tests**".

3º A continuación se mostrará una nueva ventana donde comenzarán los tests de evaluación. El usuario debe responder a cada test con el valor "**Yes**" o "**No**" para poder seguir avanzando mediante el botón "**Next >**" hasta llegar a una ventana con los resultados de los tests de evaluación. En cualquier momento se pueden comenzar de nuevo los tests de evaluación mediante el botón "< **Start tests again**".

Information Technology Governance Solver - ITGS

File Menu Options Help

1/6

< Start tests again Next >

Is strategic IT planning being performed? (Select "No" if there is no long term planning for IT. Select "Yes" if management have a strategic plan for IT.)

☐ Yes ☐ No

Is there awareness amongst management that strategic IT planning is needed to support business goals? (Select "No" if strategic IT planning would benefit the organisation, but is not performed. Select "Yes" if there is some awareness of the need for IT to be focused on supporting the achievement of business goals.)

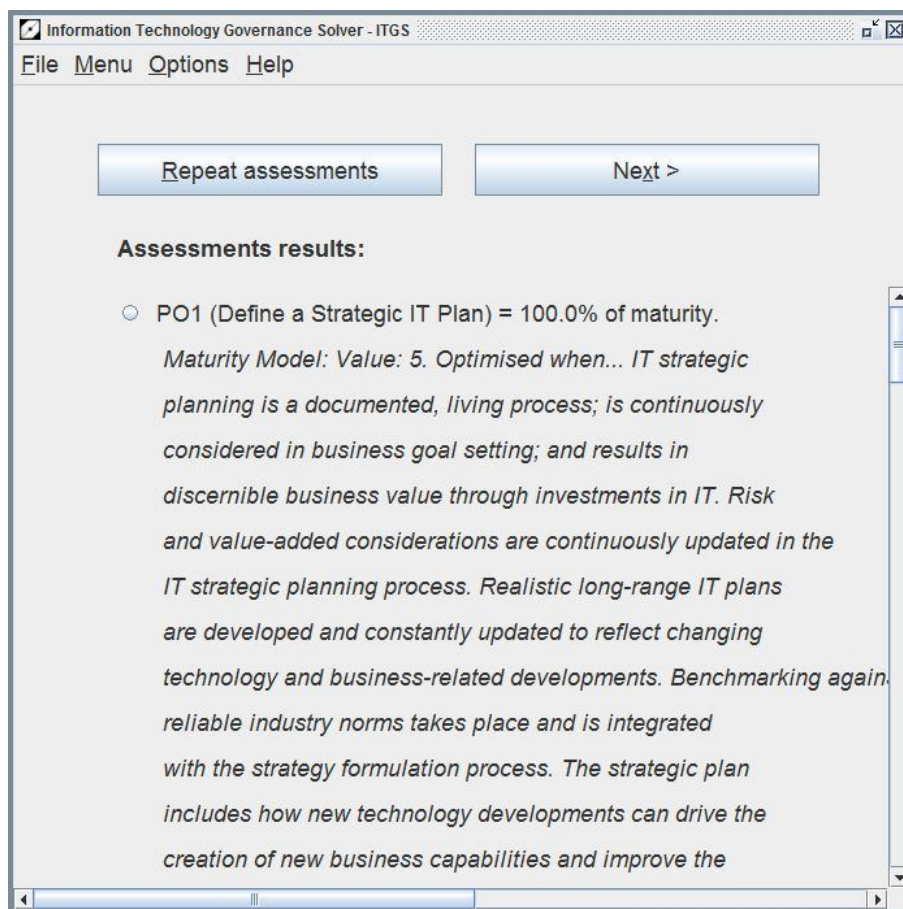
☐ Yes ☐ No

Do IT risk assessments for IT processes and business decisions occur? (Select "Yes" if some form of IT risk assessment is undertaken from time to time.)

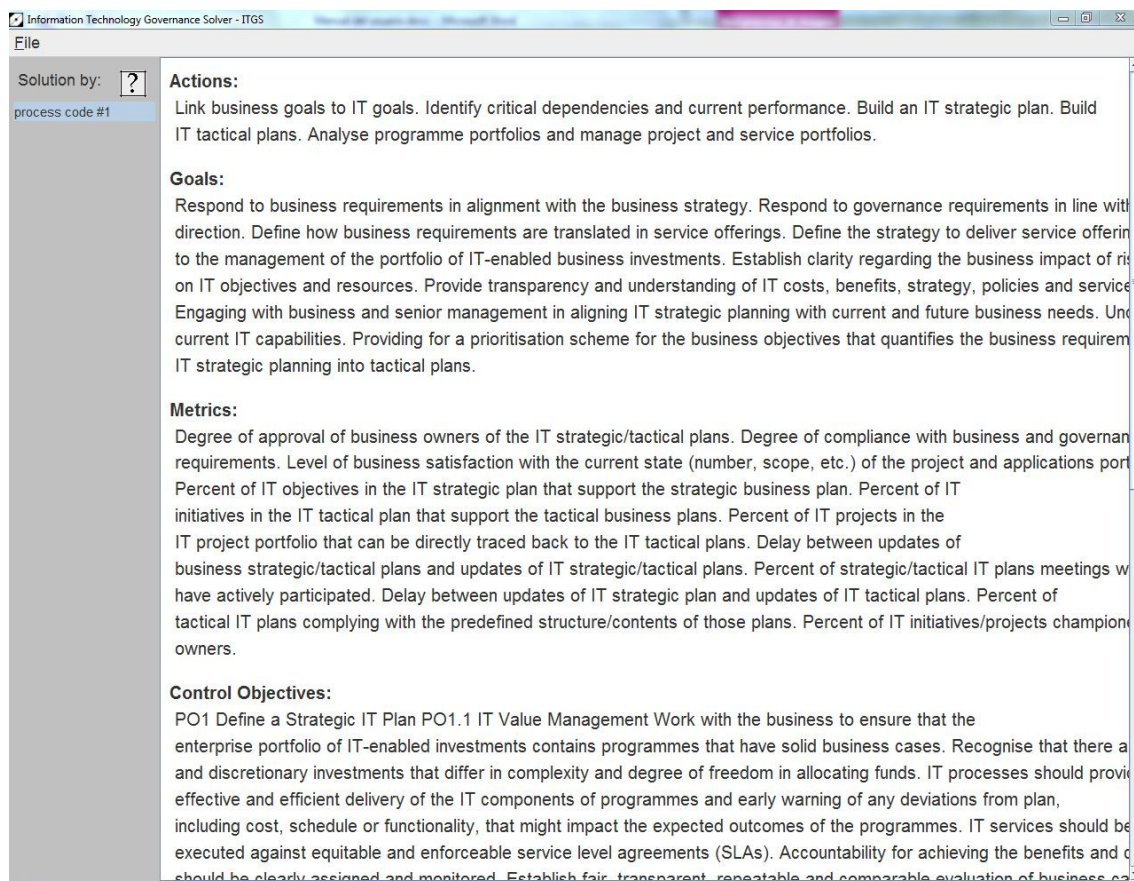
☐ Yes ☐ No

Durante la realización de los tests de evaluación el usuario puede visualizar en la parte superior de la ventana del programa, el número de ventanas que restan de tests de evaluación, y la ventana actual en la que se encuentra de tests de evaluación. En el caso del ejemplo está situado en la ventana de tests de evaluación número 1, de las 6 totales que contienen tests de evaluación.

4º Al finalizar el test de evaluación, se mostrará una nueva ventana con los resultados obtenidos, en base a los procesos de **COBIT 4.1** que están relacionados con el test.



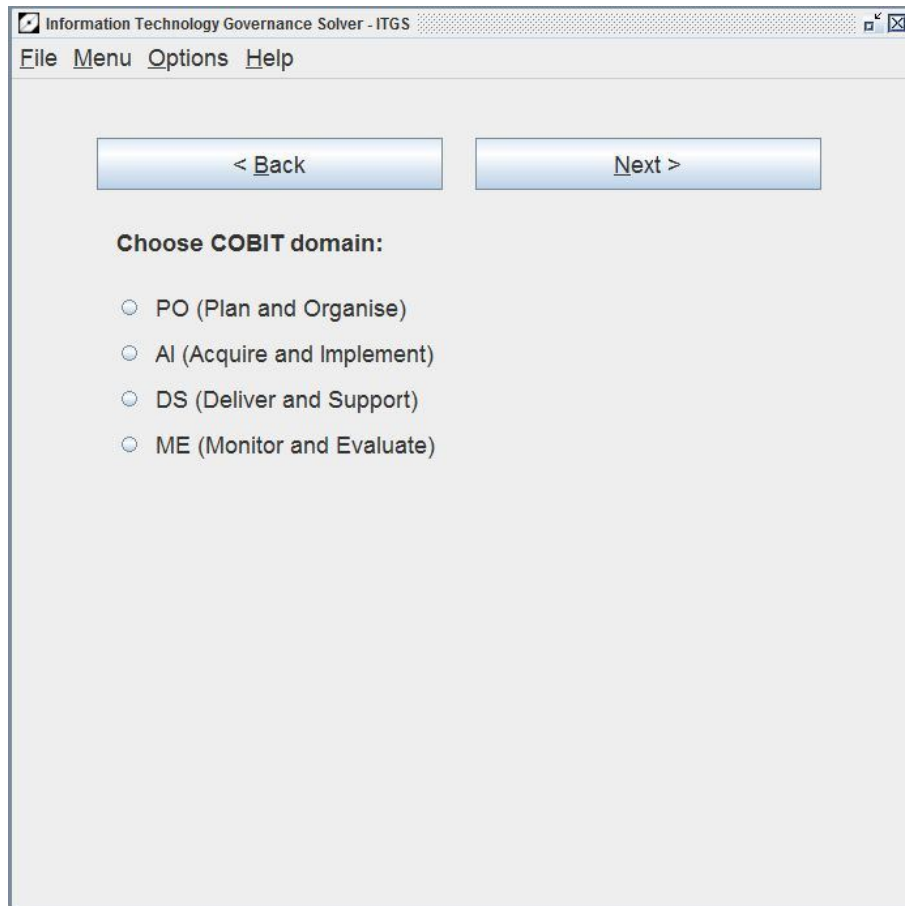
5º Seleccionando cualquiera de los procesos mostrados en la ventana anterior y haciendo clic en "**Next >**", se mostrará toda su información en una nueva ventana idéntica a la de la solución en la rama de consulta de los problemas, pudiendo también almacenar la solución en formato txt, PDF, y HTML, así como imprimir la primera página de la misma como demostración. Además, en otra ventana, el programa permite seleccionar otros procesos que han sido también evaluados, y los problemas relacionados con el proceso seleccionado anteriormente en los resultados de la evaluación, o navegar hasta la pantalla principal.



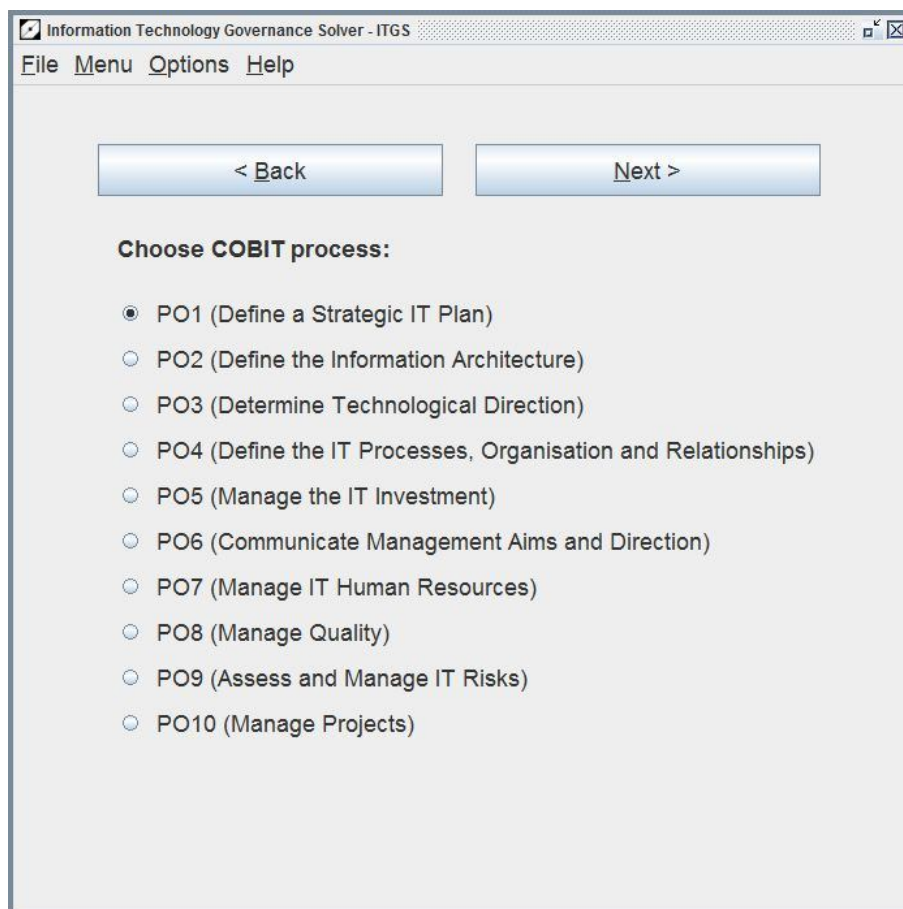
Consulta de procesos *COBIT 4.1* a través de dominios *COBIT 4.1*

Para realizar esta consulta, hay que seguir los siguientes pasos:

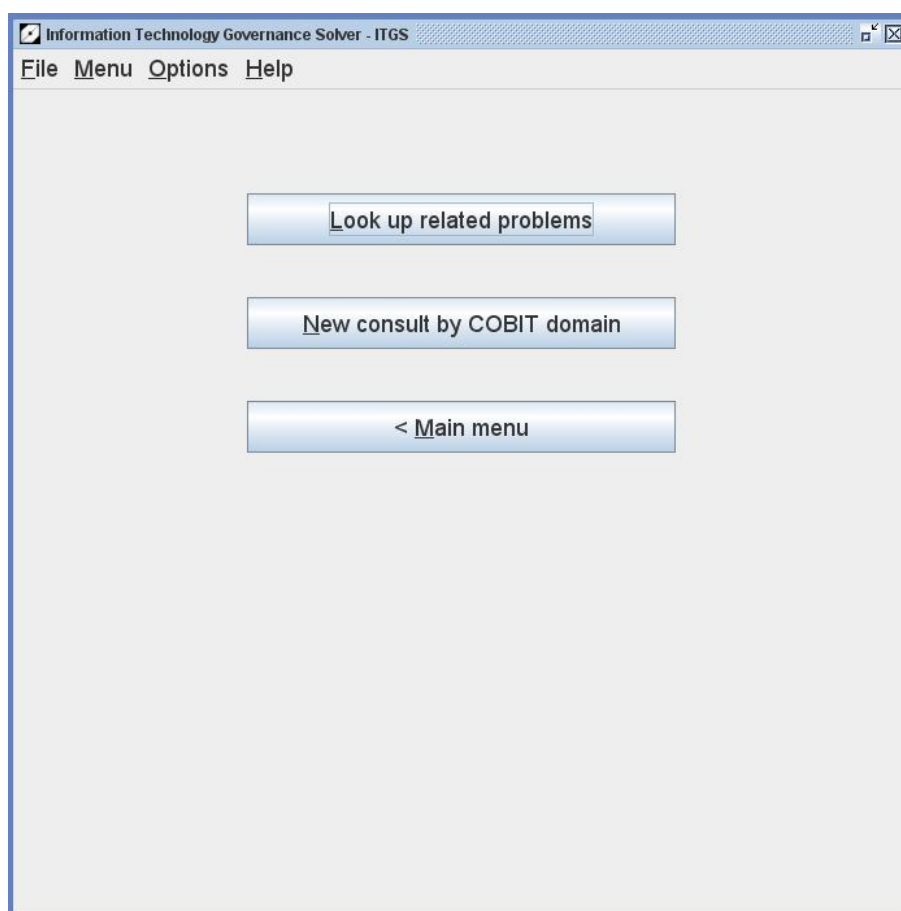
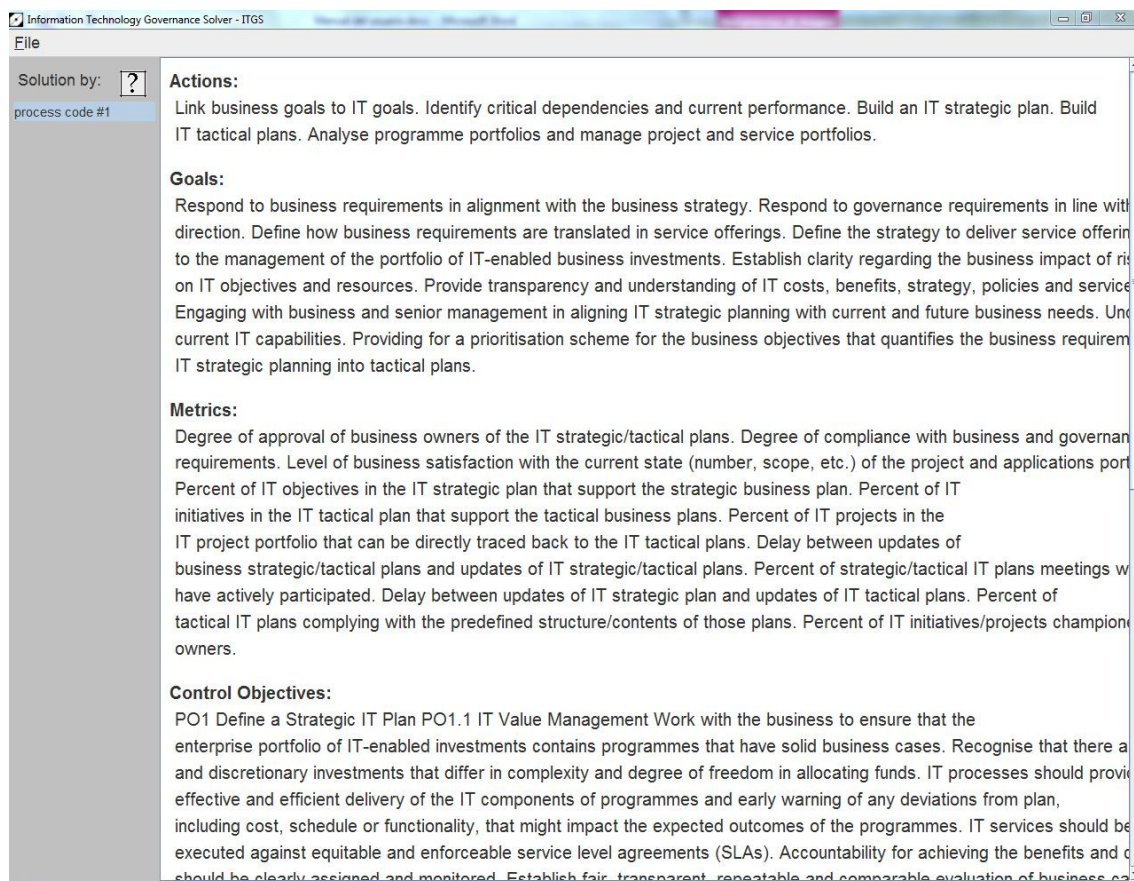
- 1º En la pantalla principal del programa, hacer clic en el botón "**User**".
- 2º Hacer clic en el botón "**Choose by COBIT domain**".
- 3º Se mostrará una ventana en la que aparecerán todos los dominios de *COBIT 4.1*.



4º Seleccionamos un dominio y hacemos clic en "**Next >**". Se muestra una nueva ventana con todos los procesos **COBIT 4.1** correspondientes al dominio seleccionado.



5º Seleccionamos un proceso y hacemos clic en "**Next >**". Se mostrará toda su información en una nueva ventana idéntica a la de la solución en la rama de consulta de los problemas, pudiendo también almacenar la solución en formato txt, PDF, y HTML, así como imprimir la primera página de la misma como demostración. Además, en otra ventana, el programa permite seleccionar otros dominios junto con sus procesos asociados, y los problemas relacionados con el proceso seleccionado anteriormente, o navegar hasta la pantalla principal.



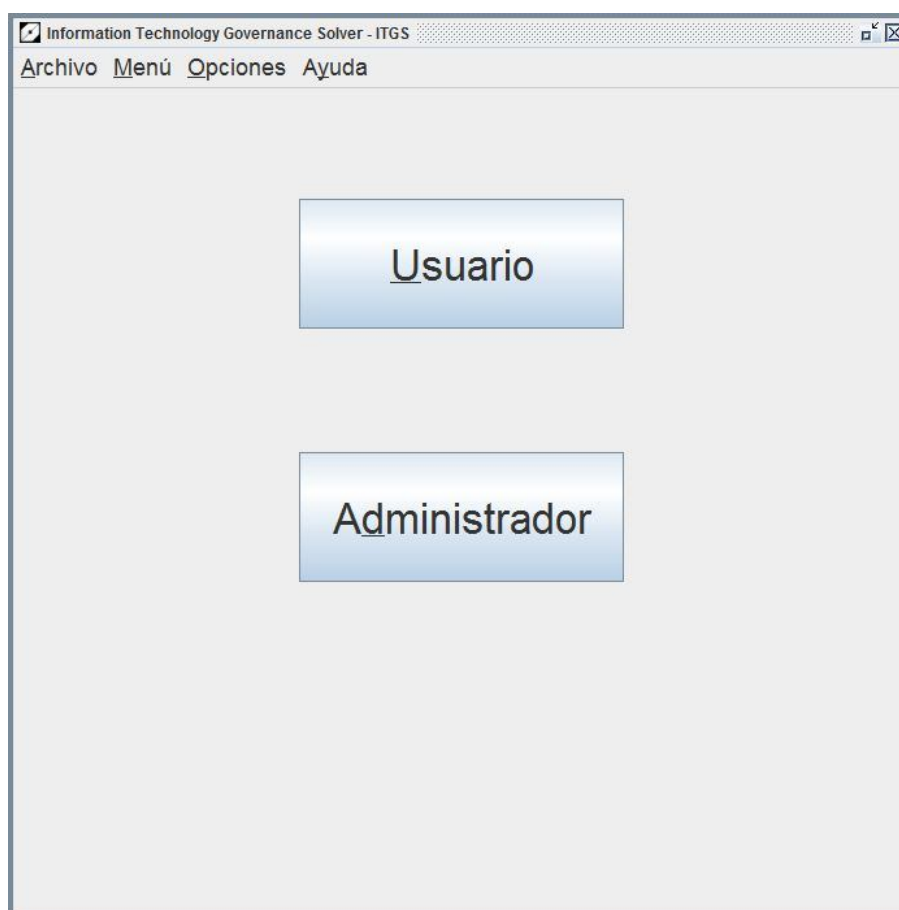
Opciones de la interfaz gráfica:

Mediante estas opciones podemos cambiar la apariencia de la interfaz gráfica, así como el idioma de la misma.

Modificación del idioma:

1º Para cambiar el idioma, desde cualquier ventana del programa, excepto la ventana de la solución, hacemos clic en el menú "**Options**" -> "**Language**" de la barra de herramientas superior.

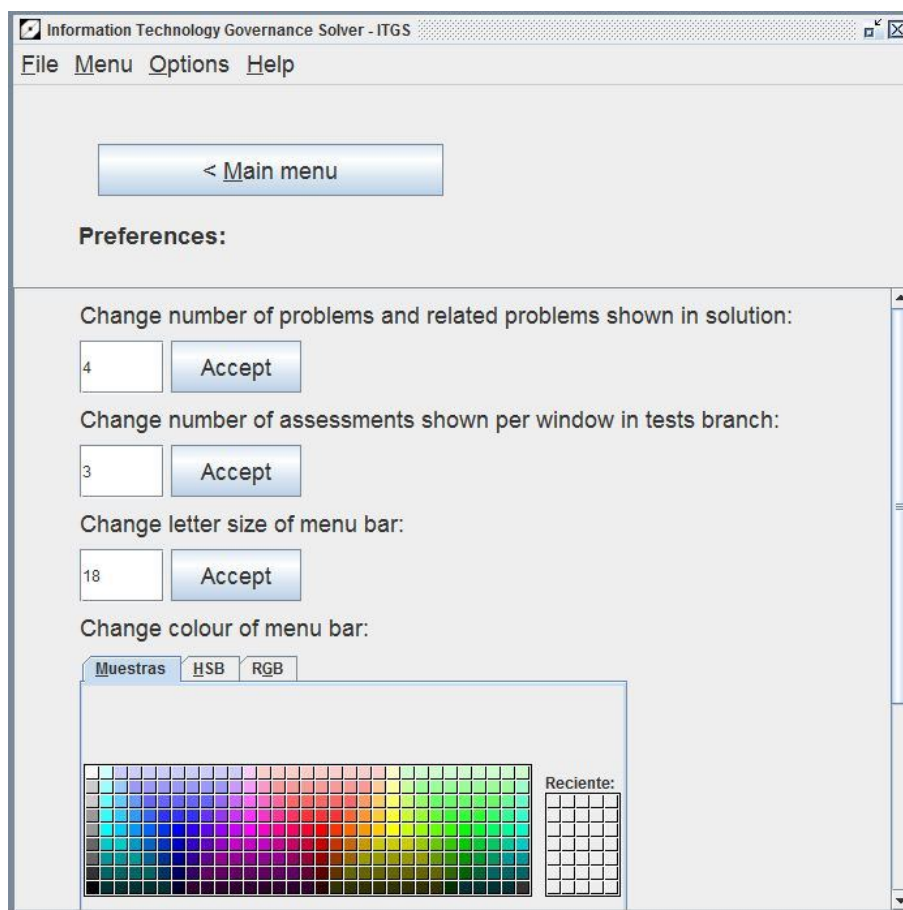
2º A continuación seleccionamos el idioma inglés –por defecto-, o español (no implementado de forma completa en esta versión). Aparecerá una ventana emergente indicando el cambio, y el programa mostrará la ventana principal en el idioma seleccionado.



Modificación de la apariencia de la interfaz gráfica:

1º Para cambiar la apariencia de la interfaz gráfica, desde cualquier ventana del programa, excepto la ventana de la solución, hacemos clic en el menú "**Options**" -> "**Preferences**" de la barra de herramientas superior.

2º Aparecerá una nueva ventana en la que se pueden modificar los siguientes parámetros:



- **Change number of problems and related problems shown in solution:** Modificación del número de problemas mostrados por ventana al realizar la búsqueda de problemas. Intervalo permitido [0, número de problemas insertados en la Base de Datos].
- **Change number of assessments shown per Windows in tests branch:** Modificación del número de evaluaciones mostradas por ventana al realizar tests de evaluación. Intervalo permitido [0, número de tests de evaluación insertados en la Base de Datos].
- **Change letter size of menú bar:** Modificación del tamaño de las letras de la barra de menú superior –en puntos–.
- **Change color of menú bar:** Modificación del color de la barra de menú superior – en formato RGB–.

- **Change by default values** (desplazando la barra lateral de desplazamiento hacia abajo): Aplicar los valores por defecto.

Para modificar cualquiera de estos parámetros basta con hacer clic en su correspondiente botón "**Accept**". Si todo ha ido bien aparecerá una ventana emergente confirmando el cambio. En caso contrario, otra ventana emergente informará de que no se ha podido realizar éste.

Ayuda:

- Manual del usuario en idioma español o inglés (no disponible este último en esta versión) –en formato PDF-.
- Información sobre el programa.

Manual del usuario:

- **1º** Para acceder al manual del usuario, desde cualquier ventana del programa, excepto la ventana de la solución, hacemos clic en el menú **"Help"** -> **"User guide"** de la barra de herramientas superior.
- **2º** Aparecerá un documento en PDF con el manual en el idioma en el que se esté mostrando el programa. El manual en idioma inglés no está disponible en esta versión.

Información sobre el programa:

- **1º** Para acceder a la información sobre el programa, desde cualquier ventana, excepto la ventana de la solución, hacemos clic en el menú **"Help"** -> **"About"** de la barra de herramientas superior.
- **2º** Se mostrará una nueva ventana emergente con la información sobre el programa.

El perfil administrador

Desde el perfil de administrador del programa se pueden realizar las siguientes operaciones:

- **Opciones de la BBDD:**
(Contenidas y cifradas en el fichero *ConfiguracionBaseDatos.txt*).
 - Modificación del host de la BBDD.
 - Modificación del puerto de la BBDD.
 - Modificación de la cadena de la BBDD.
 - Modificación del usuario de la BBDD.
 - Modificación de la contraseña de la BBDD.
- **Modificación de la contraseña del usuario administrador.**
(Contenida y cifrada en el fichero *PasswordAdministrador.txt*).
- **Modificación de la información relacionada con los problemas de la BBDD:**
 - Añadir información:
 - Añadir problemas.
 - Añadir palabras.
 - Añadir sinónimos.
 - Eliminar información:
 - Eliminar problemas.
 - Eliminar palabras.
 - Eliminar sinónimos.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los problemas.
 - Modificar los nombres de las palabras.
 - Modificar los nombres de los sinónimos.
 - Relacionar información:
 - Relacionar/des-relacionar problemas con procesos **COBIT 4.1**.
 - Relacionar/des-relacionar palabras con problemas.
 - Relacionar/des-relacionar palabras con sinónimos.

- **Modificación de la información relacionada con el test de evaluación de la BBDD:**
 - Añadir información:
 - Añadir test de evaluación.
 - Eliminar información:
 - Eliminar test de evaluación.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los tests de evaluación.
 - Relacionar información:
 - Relacionar/des-relacionar tests de evaluación con procesos **COBIT 4.1**.
- **Modificación de la información relacionada con COBIT 4.1:**
 - Añadir información:
 - Añadir dominios.
 - Añadir procesos.
 - Añadir objetivos de control.
 - Añadir objetivos TI.
 - Añadir objetivos de negocio.
 - Añadir metas.
 - Añadir métricas.
 - Añadir acciones.
 - Añadir modelos de madurez.
 - Eliminar información:
 - Eliminar dominios.
 - Eliminar procesos.
 - Eliminar objetivos de control.
 - Eliminar objetivos TI.
 - Eliminar objetivos de negocio.
 - Eliminar metas.
 - Eliminar métricas.
 - Eliminar acciones.
 - Eliminar modelos de madurez.
 - Actualizar información:
 - Modificar información:
 - Modificar los nombres de los dominios.
 - Modificar los nombres de los procesos.
 - Modificar los nombres de los objetivos de control.
 - Modificar los nombres de los objetivos TI.
 - Modificar los nombres de los objetivos de negocio.
 - Modificar los nombres de los metas.
 - Modificar los nombres de las métricas.
 - Modificar los nombres de los acciones.

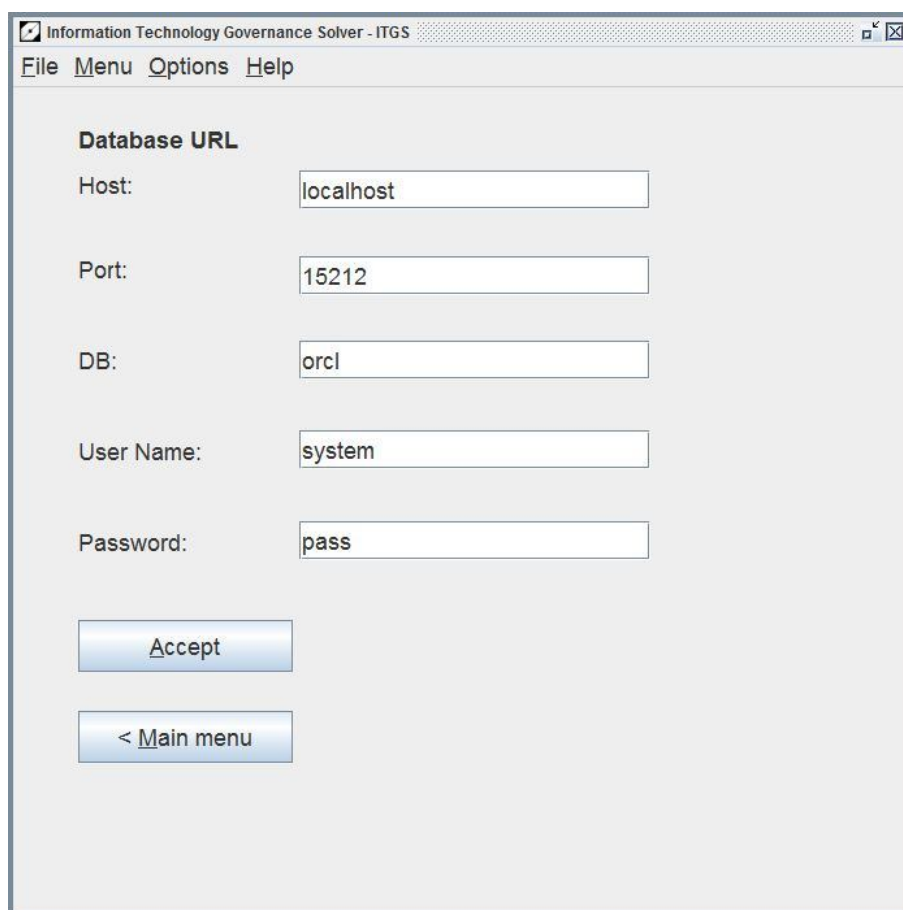
- Modificar los nombres de los modelos de madurez.
- Relacionar información:
 - Relacionar/des-relacionar dominios con procesos.
 - Relacionar/des-relacionar procesos con objetivos de control.
 - Relacionar/des-relacionar objetivos de control con objetivos TI.
 - Relacionar/des-relacionar objetivos TI con objetivos de negocio.
 - Relacionar/des-relacionar metas con métricas.
 - Relacionar/des-relacionar métricas con objetivos de control.
 - Relacionar/des-relacionar acciones con objetivos de control.
 - Relacionar/des-relacionar modelos de madurez con procesos.

Opciones de la BBDD:

(Contenidas y cifradas en el fichero *ConfiguracionBaseDatos.txt*).

Mediante estas opciones se pueden configurar todos los parámetros de conexión con la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure database parameters**".
- 4º En la nueva ventana que aparece, se podrá configurar el acceso a la base de datos de Oracle con la que se conecta el programa:



The screenshot shows a window titled "Information Technology Governance Solver - ITGS". The window has a menu bar with "File", "Menu", "Options", and "Help". The main area is titled "Database URL" and contains the following fields:

- Host: localhost
- Port: 15212
- DB: orcl
- User Name: system
- Password: pass

At the bottom of the window, there are two buttons: "Accept" and "< Main menu".

Explicación de los campos a insertar:

- **Host:** Hace referencia a la ubicación donde está instalada la base de datos. Por ejemplo, si está instalada en el equipo desde el que se utiliza el programa, su valor será *localhost*. En otro caso, se debe rellenar con la dirección IP o el nombre lógico del equipo donde esté instalada la base de datos.
- **Port:** Hace referencia al puerto desde el cual se accede a la base de datos.
- **DB:** Indica el nombre de la base de datos a la que se conectará el programa.
- **User Name:** Indica el nombre de usuario de la base de datos.
- **Password:** Indica la contraseña de la base de datos.

Una vez insertados todos los campos, hacer clic en "**Accept**". Si la conexión es correcta, aparecerá una nueva ventana emergente informando que se ha podido establecer la conexión con la Base de Datos, y aparecerá la ventana anterior con todas las opciones disponibles del usuario administrador. En caso contrario, aparecerá una ventana emergente informando de que no se ha podido establecer la conexión con la Base de Datos, y se le pedirá al usuario administrador que compruebe los parámetros de conexión.

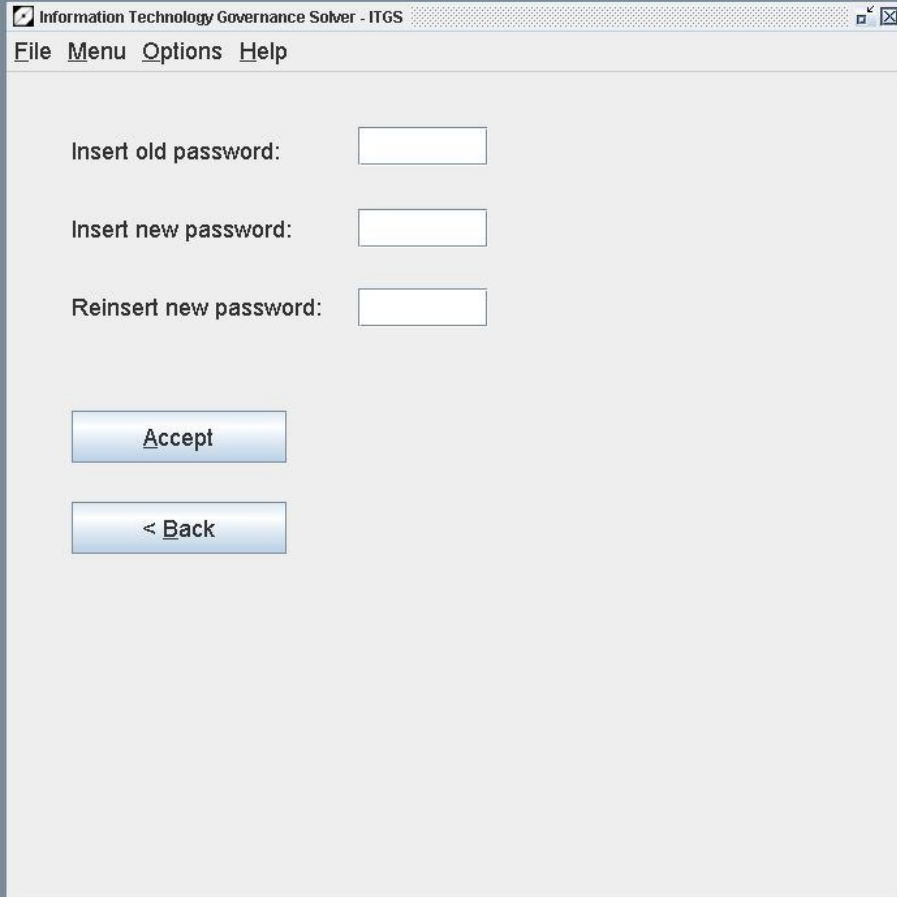
Modificación de la contraseña del usuario administrador:

(Contenida y cifrada en el fichero PasswordAdministrador.txt).

Mediante esta opción se permite el cambio de la contraseña del usuario administrador. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Change administrator password**".

4º En la nueva ventana que aparece, se podrá configurar el acceso a la base de datos de Oracle con la que se conecta el programa:



The screenshot shows a window titled "Information Technology Governance Solver - ITGS". The window has a menu bar with "File", "Menu", "Options", and "Help". The main area contains three text input fields with labels: "Insert old password:", "Insert new password:", and "Reinsert new password:". Below the fields are two buttons: "Accept" and "< Back".

5º Se debe insertar primero el la contraseña antigua en el campo "**Insert old password**", y la nueva contraseña, en los campos "**Insert new password**" y "**Reinsert new password**".

6º Hacer clic en "**Accept**", y si todo es correcto se mostrará una nueva ventana emergente informando de que se ha cambiado correctamente la contraseña, y aparecerá la ventana anterior con todas las opciones disponibles del usuario administrador. Si hay algún error, se mostrará una ventana emergente informando.

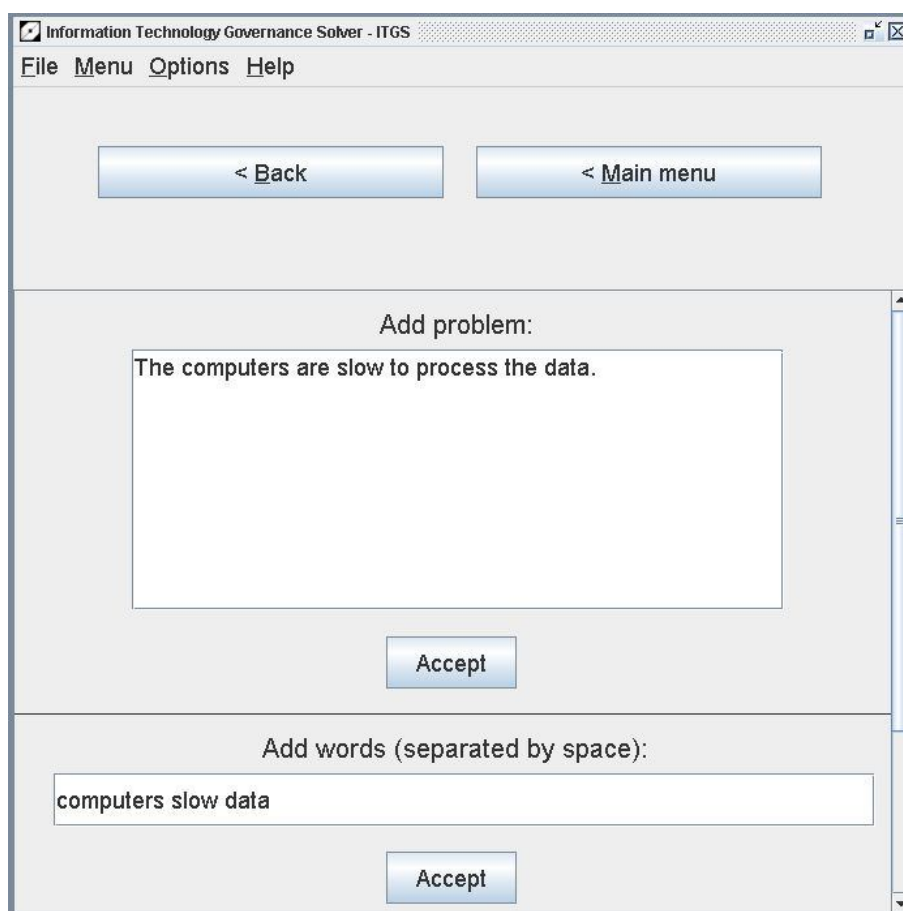
Modificación de la información relacionada con los problemas de la BBDD:

Añadir información:

Mediante esta opción el programa permite añadir nuevos problemas, palabras, y sinónimos a la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure problems information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Add information**".
- 5º Se mostrará una nueva ventana, donde se podrán añadir problemas, palabras separadas por espacios, y sinónimos separados por espacios. Una vez escritos, para añadirlos basta con hacer clic en el botón "**Accept**" de cada apartado.

- **Add problem:** Añadir problema.
- **Add words (separated by space):** Añadir palabras separadas por espacios.
- **Add synonyms (separated by space):** Añadir sinónimos separados por espacios.

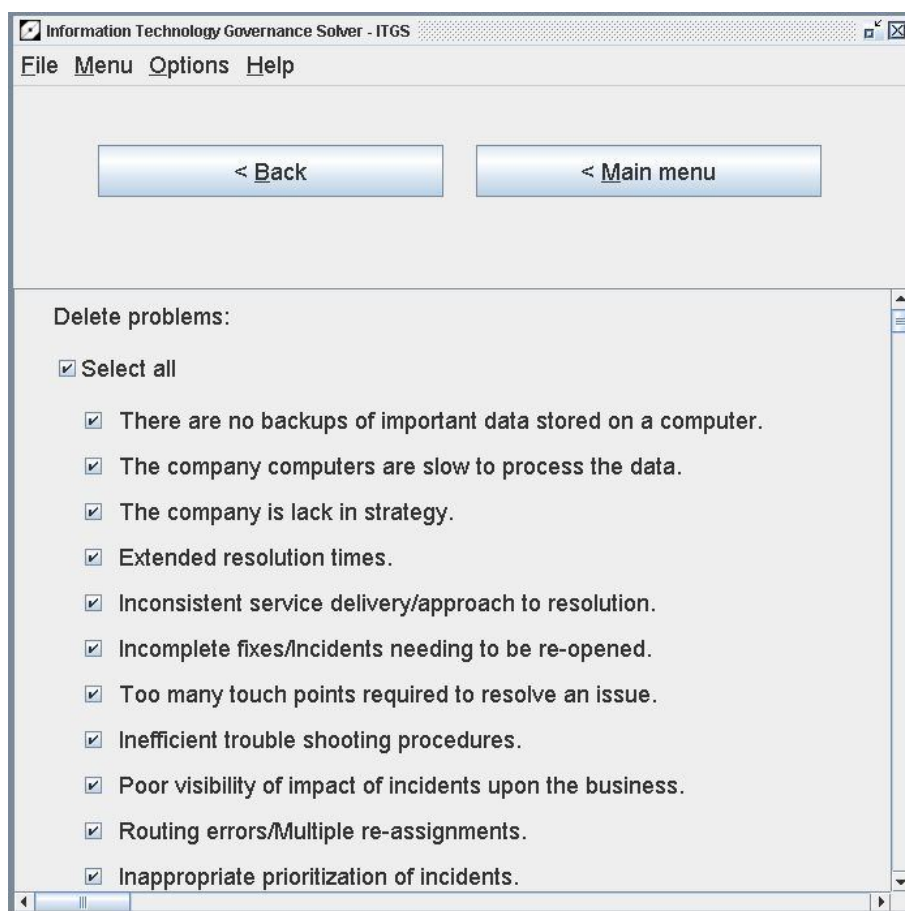


Eliminar información:

Mediante esta opción el programa permite eliminar nuevos problemas, palabras, y sinónimos de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure problems information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Delete information**".
- 5º Se mostrará una nueva ventana, donde se podrán eliminar problemas, palabras, y sinónimos. Se pueden seleccionar todos los problemas, palabras o sinónimos, haciendo clic en "**Select all**", o seleccionar manualmente. Para eliminar, basta con hacer clic en "**Accept**".

- **Delete problems:** Eliminar problemas.
- **Delete words:** Eliminar palabras.
- **Delete synonyms:** Eliminar sinónimos.



Actualizar información:

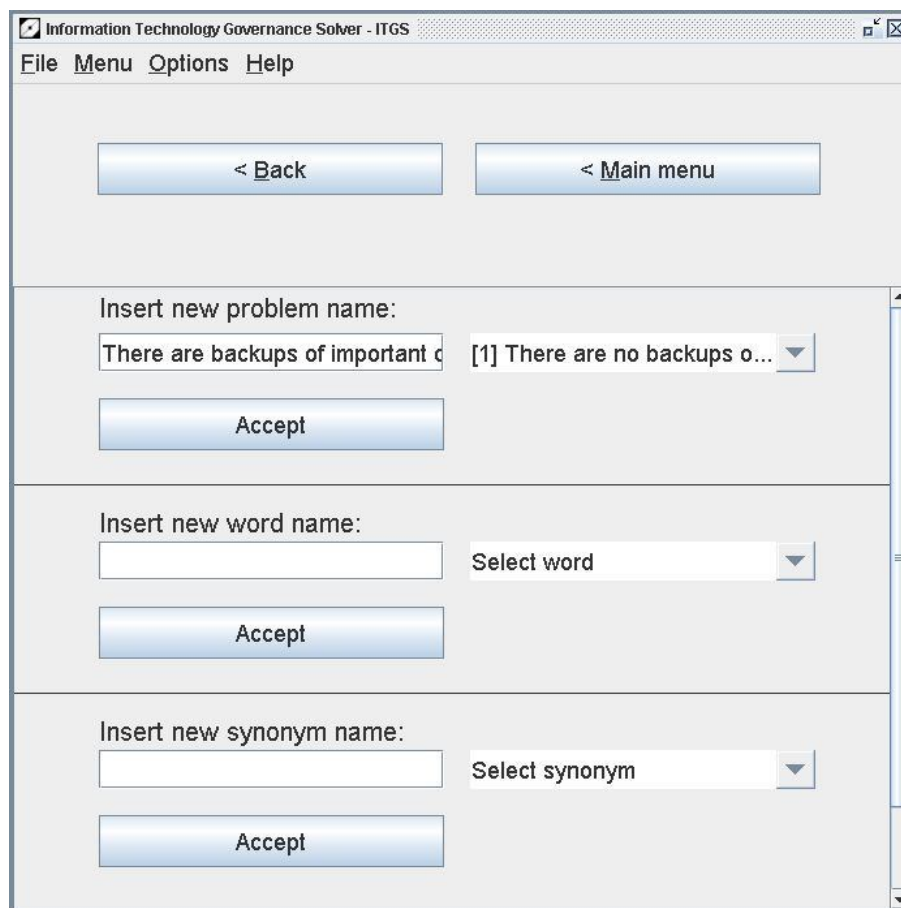
Modificar información:

Mediante esta opción el programa permite modificar los nombres de los problemas, palabras, y sinónimos de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure problems information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Update information**".
- 5º En la nueva ventana que aparece, hacer clic en "**Change names**".

6º Se mostrará una nueva ventana, donde se podrá modificar los nombres de los problemas, palabras, y sinónimos. Se selecciona la información a la que se le quiera cambiar el nombre de los menús desplegables. Se escribe el nuevo nombre en el cuadro de texto correspondiente, y por último se hace clic en **"Accept"**.

- **Insert new problema name:** Modifica el nombre de un problema.
- **Insert new word name:** Modifica el nombre de un problema.
- **Insert new synonym name:** Modifica el nombre de un problema.



Relacionar información:

Mediante esta opción el programa permite relacionar los problemas, procesos, palabras, y sinónimos de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón **"Administrator"**.
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en **"Accept"**.
- 3º Se mostrará una ventana. Hacer clic en **"Configure problems information"**.

4º En la nueva ventana que aparece, hacer clic en "**Update information**".

5º En la nueva ventana que aparece, hacer clic en "**Relate database problems tables**".

6º Se mostrará una nueva ventana, donde se podrán realizar los siguientes cambios:

- **Problem – Process relationship:** Relaciona/des-relaciona un problema con un proceso **COBIT 4.1**.
- **Word – Problem relationship:** Relaciona/des-relaciona una palabra con un problema.
- **Word – Synonym relationship:** Relaciona/des-relaciona una palabra con un sinónimo.

7º Hacer clic en "**Create relationship**" para crear una nueva relación, y en "**Delete relationship**" para eliminar una relación ya existente en la Base de Datos.

The screenshot shows the 'Information Technology Governance Solver - ITGS' application window. It has a menu bar with 'File', 'Menu', 'Options', and 'Help'. Below the menu bar are two buttons: '< Back' and '< Main menu'. The main content area is divided into three sections:

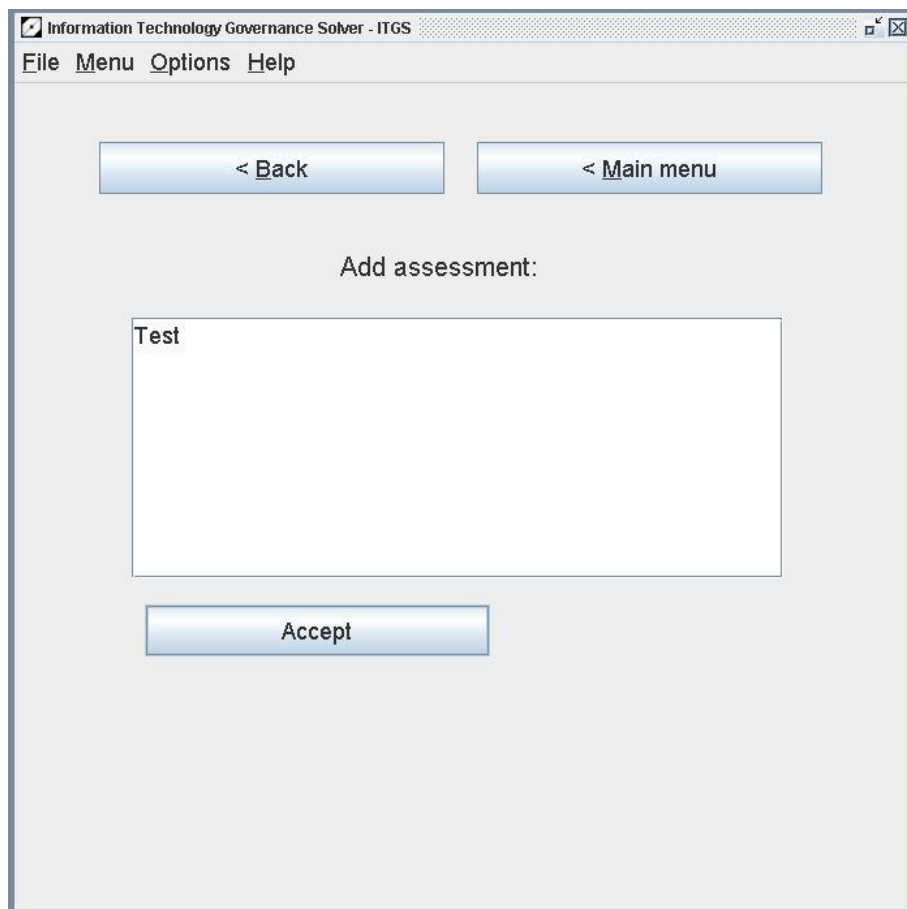
- Problem - Process relationship:** This section contains two dropdown menus. The first is labeled '[1] There are no backups o...' and the second is labeled '[28] DS11 (Manage Data)'. Below these are two buttons: 'Create relationship' and 'Delete relationship'.
- Word - Problem relationship:** This section contains two dropdown menus labeled 'Select word' and 'Select problem'. Below these are two buttons: 'Create relationship' and 'Delete relationship'.
- Word - Synonym relationship:** This section contains two dropdown menus labeled 'Select word' and 'Select synonym'. Below these are two buttons: 'Create relationship' and 'Delete relationship'.

Modificación de la información relacionada con los tests de evaluación de la BBDD:

Añadir información:

Mediante esta opción el programa permite añadir nuevos tests de evaluación a la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
 - 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
 - 3º Se mostrará una ventana. Hacer clic en "**Configure assessments information**".
 - 4º En la nueva ventana que aparece, hacer clic en "**Add information**".
 - 5º Se mostrará una nueva ventana, donde se podrán añadir tests de evaluación. Una vez escritos, para añadirlos basta con hacer clic en el botón "**Accept**".
- **Add assessment:** Añadir test de evaluación.



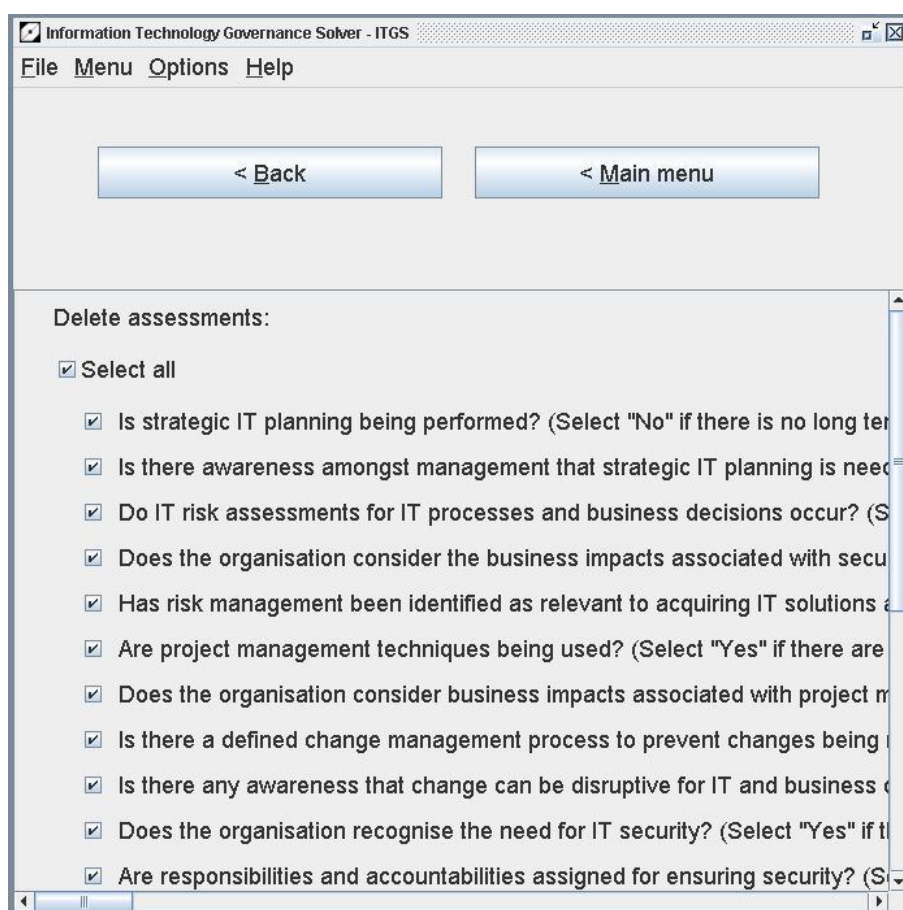
The screenshot shows a window titled "Information Technology Governance Solver - ITGS". The menu bar includes "File", "Menu", "Options", and "Help". At the top, there are two buttons: "< Back" and "< Main menu". Below these, the text "Add assessment:" is displayed. Underneath is a large text input field with the placeholder text "Test". At the bottom of the window is an "Accept" button.

Eliminar información:

Mediante esta opción el programa permite eliminar tests de evaluación de la Base de Datos.
Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure assessments information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Delete information**".
- 5º Se mostrará una nueva ventana, donde se podrán eliminar tests de evaluación. Se pueden seleccionar todos los tests de evaluación, haciendo clic en "**Select all**", o seleccionar manualmente. Para eliminar, basta con hacer clic en "**Accept**".

- **Delete assessments:** Eliminar tests de evaluación.

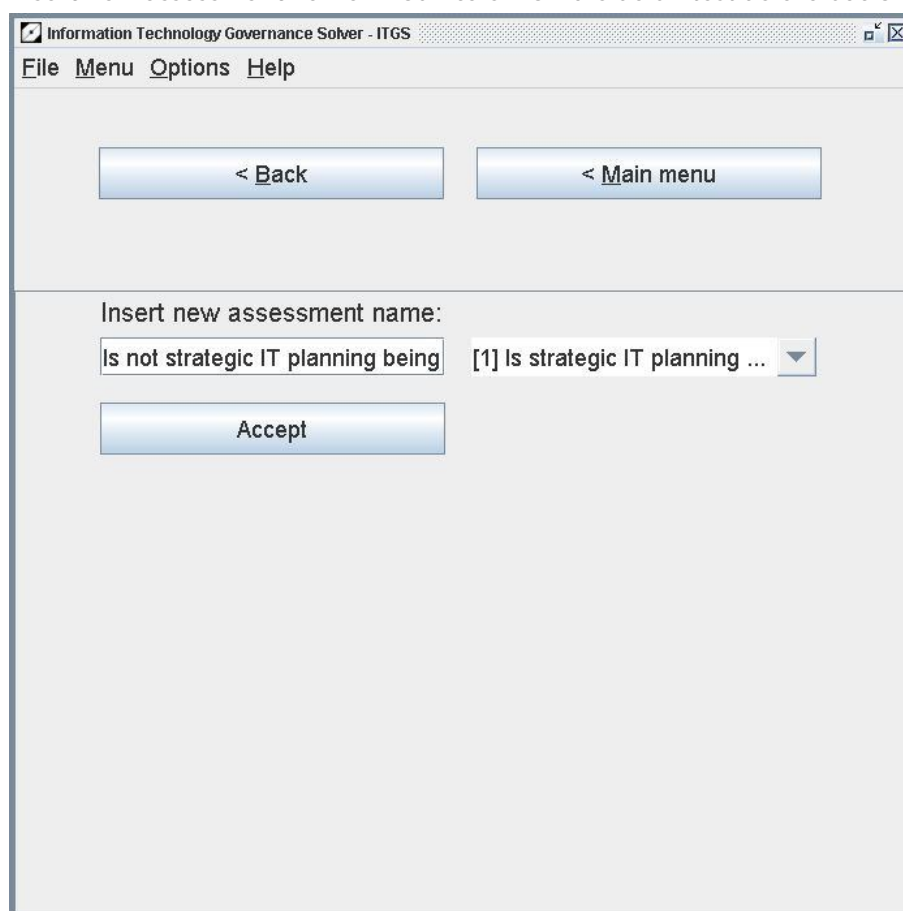


Actualizar información:**Modificar información:**

Mediante esta opción el programa permite modificar los nombres de los tests de evaluación de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón **"Administrator"**.
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en **"Accept"**.
- 3º Se mostrará una ventana. Hacer clic en **"Configure assessments information"**.
- 4º En la nueva ventana que aparece, hacer clic en **"Update information"**.
- 5º En la nueva ventana que aparece, hacer clic en **"Change names"**.
- 6º Se mostrará una nueva ventana, donde se podrán modificar los nombres de los tests de evaluación. Se selecciona la información a la que se le quiera cambiar el nombre del menú desplegable. Se escribe el nuevo nombre en el cuadro de texto correspondiente, y por último se hace clic en **"Accept"**.

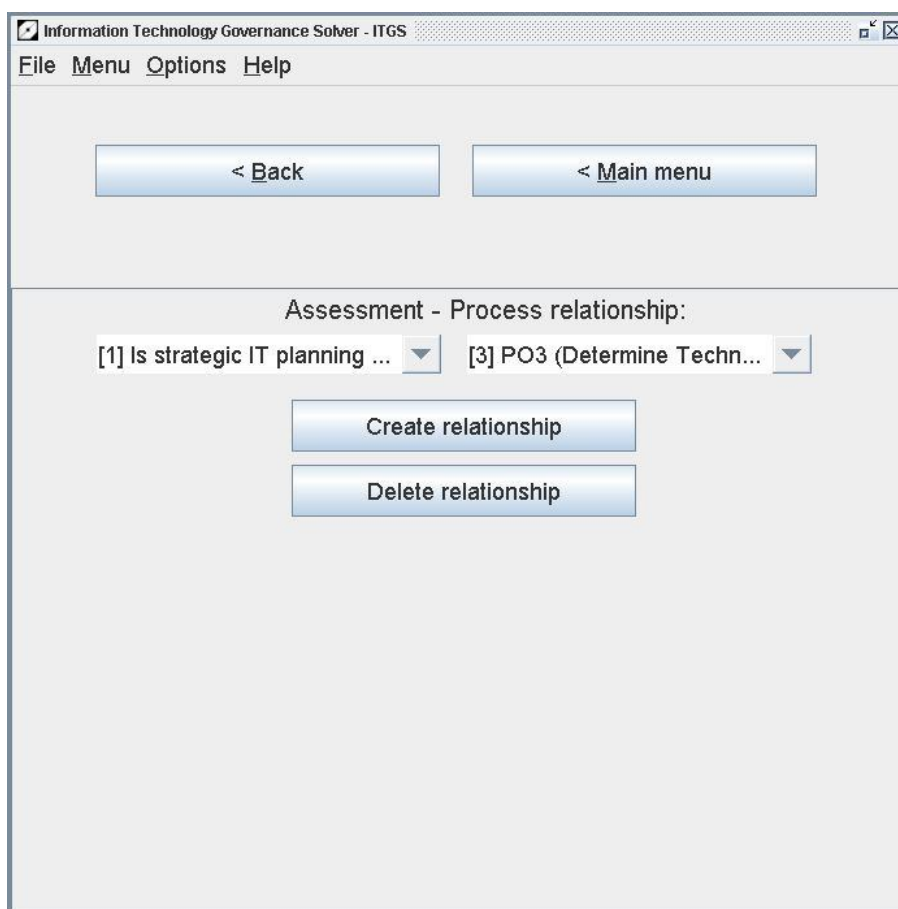
- **Insert new assessment name:** Modifica el nombre de un test de evaluación.



Relacionar información:

Mediante esta opción el programa permite relacionar los tests de evaluación y los procesos de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure assessments information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Update information**".
- 5º En la nueva ventana que aparece, hacer clic en "**Relate database assessments tables**".
- 6º Se mostrará una nueva ventana, donde se podrán realizar los siguientes cambios:
 - **Assessment – Process relationship:** Relaciona/des-relaciona un test de evaluación con un proceso **COBIT 4.1**.
- 7º Hacer clic en "**Create relationship**" para crear una nueva relación, y en "**Delete relationship**" para eliminar una relación ya existente en la Base de Datos.

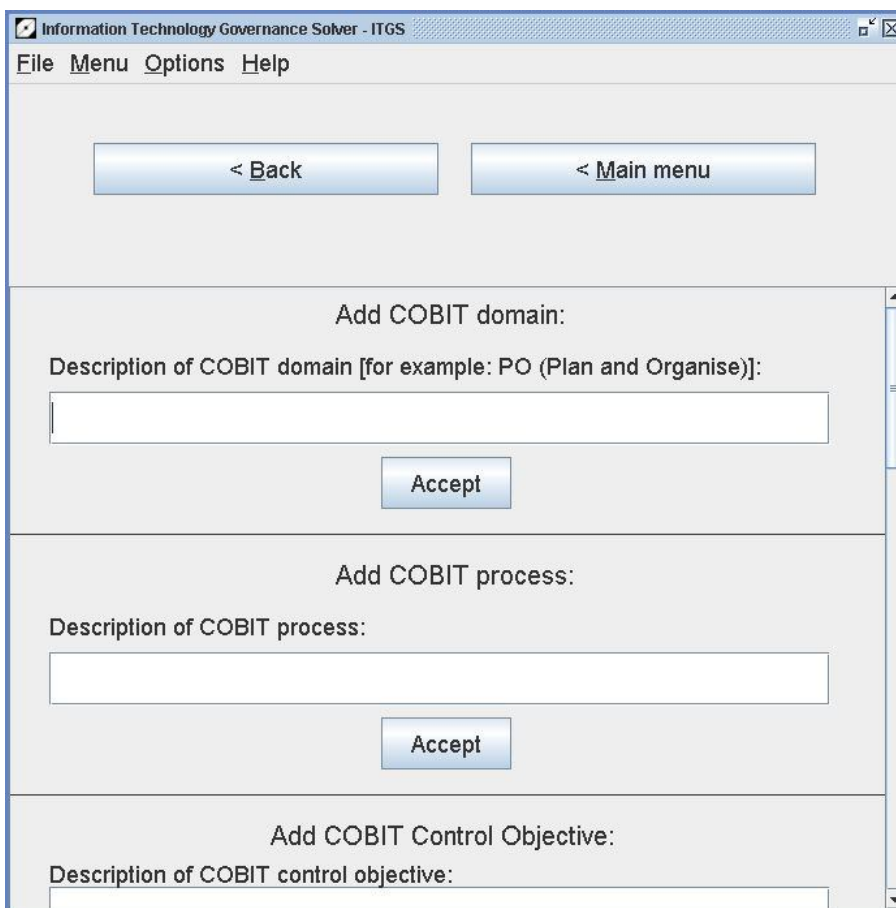


Modificación de la información relacionada con COBIT 4.1:

Añadir información:

Mediante esta opción el programa permite añadir nuevos parámetros relacionados con **COBIT 4.1** a la Base de Datos. Pasos:

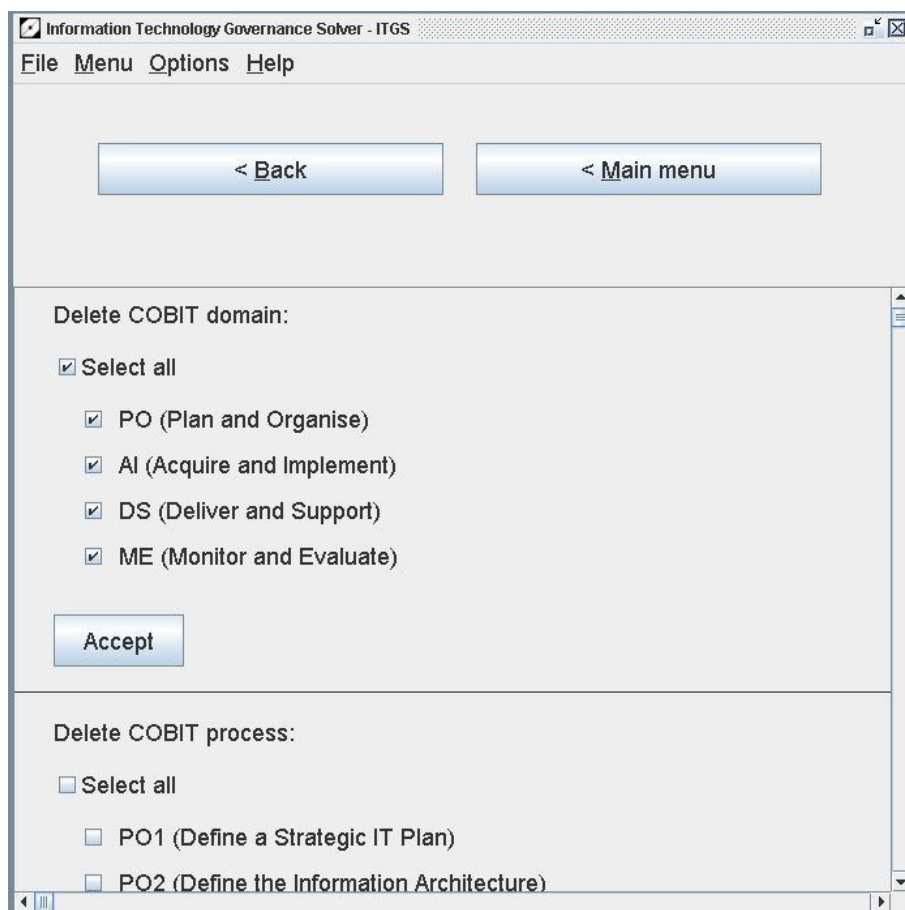
- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure COBIT information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Add information**".
- 5º Se mostrará una nueva ventana, donde se podrán añadir nuevos parámetros relacionados con **COBIT 4.1**. Una vez escritos, para añadirlos basta con hacer clic en el botón "**Accept**".



Eliminar información:

Mediante esta opción el programa permite eliminar parámetros relacionados con **COBIT 4.1** de la Base de Datos. Pasos:

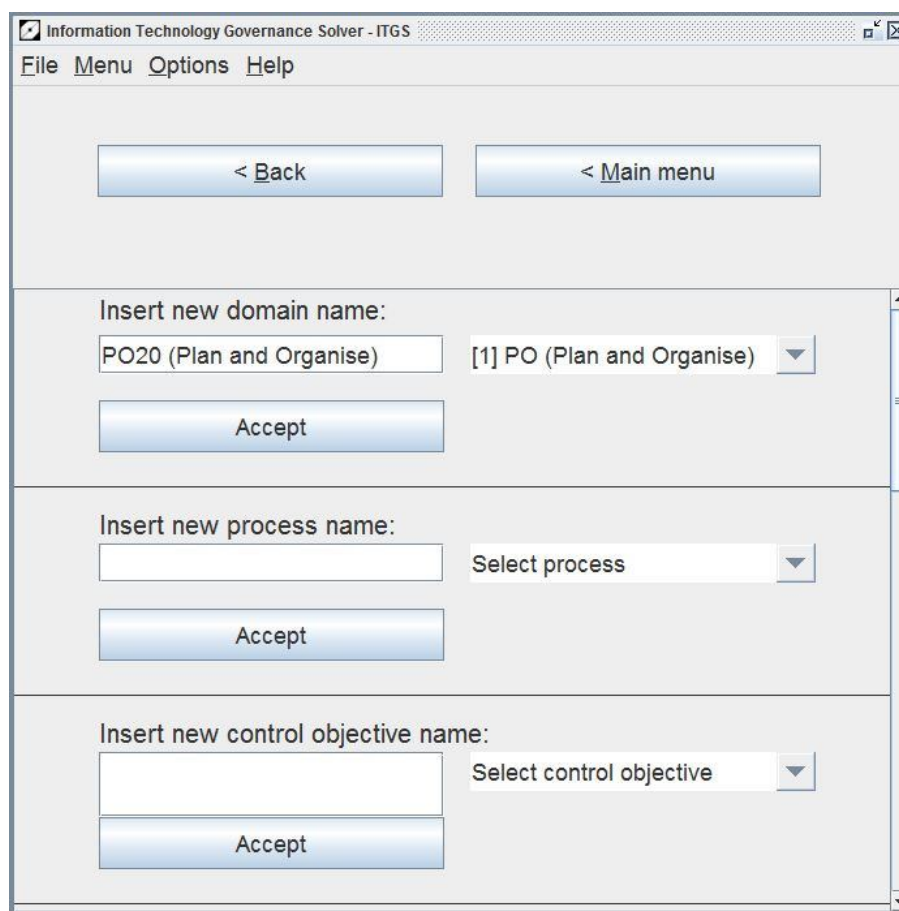
- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure COBIT information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Delete information**".
- 5º Se mostrará una nueva ventana, donde se podrán eliminar parámetros relacionados con **COBIT 4.1**. Se pueden seleccionar todos los parámetros de un apartado, haciendo clic en "**Select all**", o seleccionar manualmente. Para eliminar, basta con hacer clic en "**Accept**".



Actualizar información:**Modificar información:**

Mediante esta opción el programa permite modificar los nombres de los parámetros relacionados con **COBIT 4.1** de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure COBIT information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Update information**".
- 5º En la nueva ventana que aparece, hacer clic en "**Change names**".
- 6º Se mostrará una nueva ventana, donde se podrán modificar los nombres de los parámetros relacionados con **COBIT 4.1**. Se selecciona la información a la que se le quiera cambiar el nombre del menú desplegable. Se escribe el nuevo nombre en el cuadro de texto correspondiente, y por último se hace clic en "**Accept**".



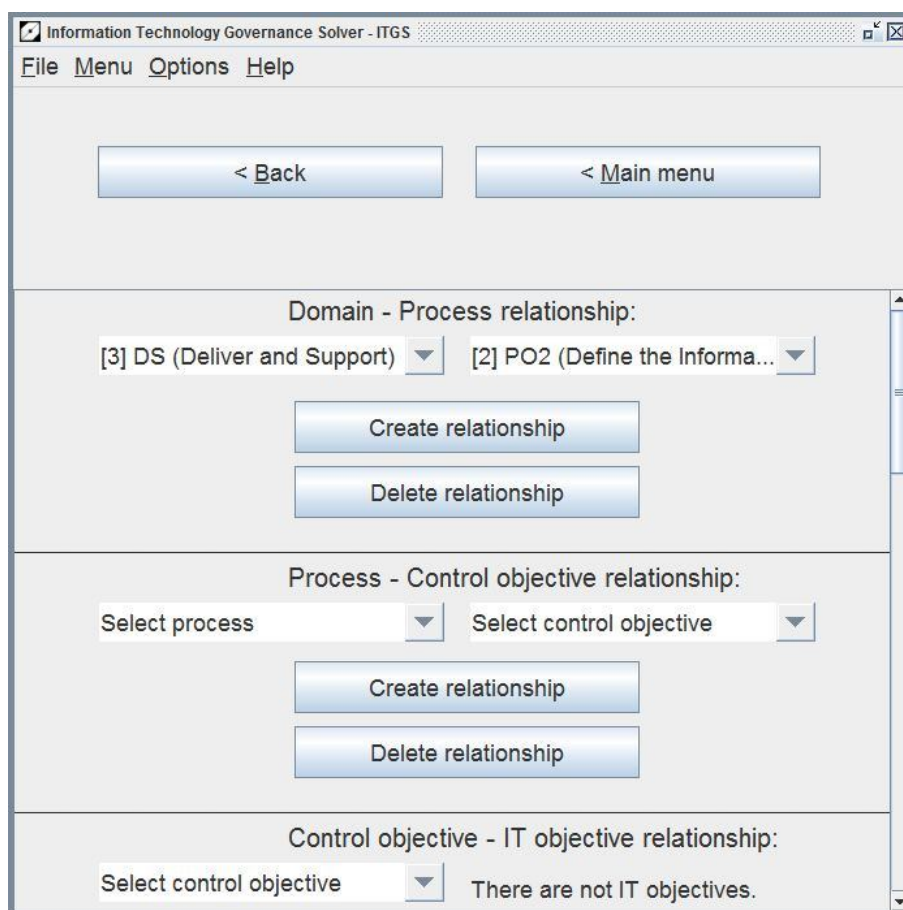
The screenshot shows a window titled "Information Technology Governance Solver - ITGS". The menu bar includes "File", "Menu", "Options", and "Help". At the top, there are two buttons: "< Back" and "< Main menu". The window is divided into three sections for editing COBIT 4.1 parameters:

- Insert new domain name:** A text box contains "PO20 (Plan and Organise)" and a dropdown menu shows "[1] PO (Plan and Organise)". An "Accept" button is below.
- Insert new process name:** A text box is empty, and a dropdown menu shows "Select process". An "Accept" button is below.
- Insert new control objective name:** A text box is empty, and a dropdown menu shows "Select control objective". An "Accept" button is below.

Relacionar información:

Mediante esta opción el programa permite relacionar los parámetros de **COBIT 4.1** de la Base de Datos. Pasos:

- 1º En la pantalla principal del programa, hacer clic en el botón "**Administrator**".
- 2º En la nueva ventana que aparece, introducir el password de usuario administrador (por defecto *password*), y hacer clic en "**Accept**".
- 3º Se mostrará una ventana. Hacer clic en "**Configure COBIT information**".
- 4º En la nueva ventana que aparece, hacer clic en "**Update information**".
- 5º En la nueva ventana que aparece, hacer clic en "**Relate database COBIT tables**".
- 6º Se mostrará una nueva ventana, donde se podrán relacionar los parámetros de **COBIT 4.1**.
- 7º Hacer clic en "**Create relationship**" para crear una nueva relación, y en "**Delete relationship**" para eliminar una relación ya existente en la Base de Datos.



Resolución de problemas

- No se puede ejecutar el programa. -> Instalar el entorno de ejecución de Java.
- No se puede conectar con la Base de Datos. -> Configurar la Base de Datos Oracle.
- No se puede acceder al menú del usuario administrador. -> Eliminar el fichero que contiene la contraseña del usuario administrador, y volver a ejecutar de nuevo el programa. Se restablecerá la contraseña por defecto: *"password"*.

Copyright

Copyright © 2012 Universidad Carlos III de Madrid. Todos los derechos reservados.